BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

L Number	Hits	Search Text	DB	Time stamp
1	1	6094647.pn.	USPAT; US-PGPUB; JPO	2004/03/29 10:18
2	2	("6487641" or "6094647").pn.	USPAT; US-PGPUB; JPO	2004/03/29 10:18
3	372	retriev$ same process same perform$ same decid$	USPAT; US-PGPUB; JPO	2004/03/29 10:41
4	15	707/3.ccls. and (retriev$ same process same perform$ same decid$)	USPAT; US-PGPUB; JPO	2004/03/29 10:41
5	10	707/4.ccls. and (retriev$ same process same perform$ same decid$)	USPAT; US-PGPUB; JPO	2004/03/29 10:41
6	1	707/6.ccls. and (retriev$ same process same perform$ same decid$)	USPAT; US-PGPUB; JPO	2004/03/29 10:42
7	8	707/2.ccls. and (retriev$ same process same perform$ same decid$)	USPAT; US-PGPUB; JPO	2004/03/29 10:42
-	199819	pattern same retriev$ same network samne conditionsame search	USPAT; US-PGPUB; JPO	2004/03/24 11:30
-	2	pattern same retriev$ same network same condition same search	USPAT; US-PGPUB; JPO	2004/03/28 09:33
-	0	pattern same retriev$ same network same expression same search	USPAT; US-PGPUB; JPO	2004/03/24 11:31
-	6	retriev$ same network same expression same search	USPAT; US-PGPUB; JPO	2004/03/24 12:37
-	29	multi$1user same search same retriev$	USPAT; US-PGPUB; JPO	2004/03/24 11:49
-	1	multi$1termina1 same search same retriev$	USPAT; US-PGPUB; JPO	2004/03/27 10:35
-	0	multi$1device same search same retriev$	USPAT; US-PGPUB; JPO	2004/03/24 11:50
-	44	plura1 same user same search same retriev$	USPAT; US-PGPUB; JPO	2004/03/24 11:51
-	6	retriev$ same network same expression same search same network	USPAT; US-PGPUB; JPO	2004/03/24 16:24
-	13232	retriev$ same process same perform$	USPAT; US-PGPUB; JPO	2004/03/24 16:25
-	372	retriev$ same process same perform$ same decid$	USPAT; US-PGPUB; JPO	2004/03/29 10:41
-	70	retriev$ same process same perform$ same decid$ same generat$	USPAT; US-PGPUB; JPO	2004/03/24 16:29
-	22	retriev$ same process same perform$ same decid$ same generat$ same table	USPAT; US-PGPUB; JPO	2004/03/24 16:30
-	0	"determining whether" same process same "is being performed"	USPAT; US-PGPUB; JPO	2004/03/24 16:55
-	0	process same "is being performed"	USPAT; US-PGPUB; JPO	2004/03/24 16:55
-	12315	"determining whether" same process	USPAT; US-PGPUB; JPO	2004/03/24 16:55

See 6094647

-	1779	"determining whether" same process same performed	USPAT; US-PGPUB; JPO	2004/03/24 16:55
-	0	"determining whether" same process same performed same "is being"	USPAT; US-PGPUB; JPO	2004/03/24 16:56
-	0	"determining whether" same process same performed same "is"	USPAT; US-PGPUB; JPO	2004/03/24 16:56
-	1779	"determining whether" same process same performed	USPAT; US-PGPUB; JPO	2004/03/24 16:57
-	0	"determining whether" same process same performed same "is"	USPAT; US-PGPUB; JPO	2004/03/24 16:57
-	0	"determining whether" same process same performed same "has been"	USPAT; US-PGPUB; JPO	2004/03/24 16:57
-	5452	query same users	USPAT; US-PGPUB; JPO	2004/03/27 08:34
-	0	query same users same terminal$1devices	USPAT; US-PGPUB; JPO	2004/03/27 08:34
-	20	query same users same terminal same devices	USPAT; US-PGPUB; JPO	2004/03/27 08:49
-	11	retriev$ same users same perform$ same table same status	USPAT; US-PGPUB; JPO	2004/03/27 08:52
-	46	retriev$ same users same table same status	USPAT; US-PGPUB; JPO	2004/03/27 09:26
-	0	"query status" same users same table	USPAT; US-PGPUB; JPO	2004/03/27 09:27
-	61	quer$1 same status same users same table	USPAT; US-PGPUB; JPO	2004/03/27 09:27
-	8	process same control same multi$1process same status	USPAT; US-PGPUB; JPO	2004/03/27 10:47
-	0	quer$ same control same multi$1users same status	USPAT; US-PGPUB; JPO	2004/03/27 10:47
-	93	quer$ same control same users same status	USPAT; US-PGPUB; JPO	2004/03/27 10:47
-	14	quer$ same control same users same status same table	USPAT; US-PGPUB; JPO	2004/03/27 10:52
-	639	Aho-Corasick or EBM or Expanded-Boyer-Moore or Shinohara-Arikawa	USPAT; US-PGPUB; JPO	2004/03/27 10:55
-	1	Aho-Corasicksame(EBM or Expanded-Boyer-Moore)same Shinohara-Arikawa	USPAT; US-PGPUB; JPO	2004/03/27 10:54
-	6	Aho-Corasick or Expanded-Boyer-Moore or Shinohara-Arikawa	USPAT; US-PGPUB; JPO	2004/03/27 11:01
-	16	(pattern adj search) same quer$	USPAT; US-PGPUB; JPO	2004/03/27 11:02
-	0	((query or retrieval) adj expression) same ((query or retrieval) adj variable) same table	USPAT; US-PGPUB; JPO	2004/03/27 21:51
-	28	((query or retrieval) same expression) same ((query or retrieval) same variable) same table	USPAT; US-PGPUB; JPO	2004/03/27 22:38
-	1	6487641.pn.	USPAT; US-PGPUB; JPO	2004/03/27 22:38

-	13	((retriev$ or search$ or query) adj (expression and variable))	USPAT; US-PGPUB; JPO	2004/03/28 09:34
-	1	6487566.pn.	USPAT; US-PGPUB; JPO	2004/03/29 10:17

US006029165A

United States Patent [19]

Gable

[11] Patent Number: 6,029,165

[45] Date of Patent: Feb. 22, 2000

[54] **SEARCH AND RETRIEVAL INFORMATION SYSTEM AND METHOD**

[75] Inventor: **Gary A. Gable**, Port Charlotte, Fla.

[73] Assignee: **Arthur Andersen LLP**, Chicago, Ill.

[21] Appl. No.: **08/967,775**

[22] Filed: **Nov. 12, 1997**

[51] **Int. Cl.**[7] **G06F 17/30**

[52] **U.S. Cl.** **707/3**; 707/1

[58] **Field of Search** 707/3, 1, 4, 5, 707/104

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,659,742	8/1997	Beattie et al.	707/104
5,675,788	10/1997	Husick et al.	707/104
5,717,914	2/1998	Husick et al.	707/5
5,737,734	4/1998	Schultz	707/5
5,742,816	4/1998	Barr et al.	707/104
5,873,076	12/1998	Barr et al.	707/3
5,920,858	7/1999	Kitabayashi et al.	707/4

OTHER PUBLICATIONS

Dunlap, C., "Informix releases new version of Web development tool", *Information Week*, Jun. 17, 1996.

Higgins, K., "Your Agent is Calling—They may still look like Web browsers, but agents are gaining sophistication and coming your way", *Communications Week*, Aug. 5, 1996.

Jacobs, I., "A World of Choices—Oracle, Informix and Sybase offer ways to Web-enable databases", *Information Week*, Jun. 15, 1996.

Marshall, M., "Oracle Goes Modular—'Cartridges' ease development of custom applications", *Information Week*, Sep. 30, 1996.

Rodriguez, K., "Excalibur Advances retrieval", *Communications Week*, Nov. 6, 1995.

Rodriguez, K., "Searching the Internet—Demand for Internet file retrieval fuels growth of search engine market", *Interactive Age*, Jul. 31, 1995.

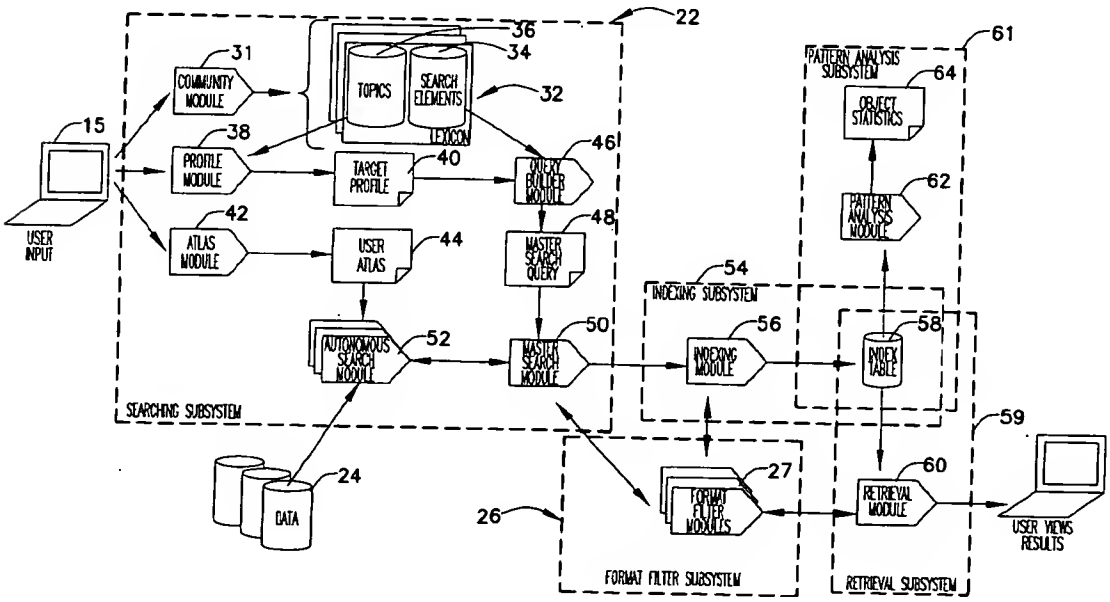
Smith, T., "'MERGER': Informix, Illustra team up—Universal Database On Its Way", *Information Week*, Feb. 19, 1996.

Primary Examiner—Paul R. Lintz
Assistant Examiner—Sanjiv Shah
Attorney, Agent, or Firm—Merchant & Gould P.C.

[57] **ABSTRACT**

A system and method for search and retrieval of electronic objects, the objects including electronically encoded information. The system and method use an electronic lexicon which is configured to provide predefined search elements that are designed to identify objects relevant to a specific community. Format filter modules identify a format of an electronic object to be searched and enable the search using the search elements within the lexicon.

30 Claims, 12 Drawing Sheets



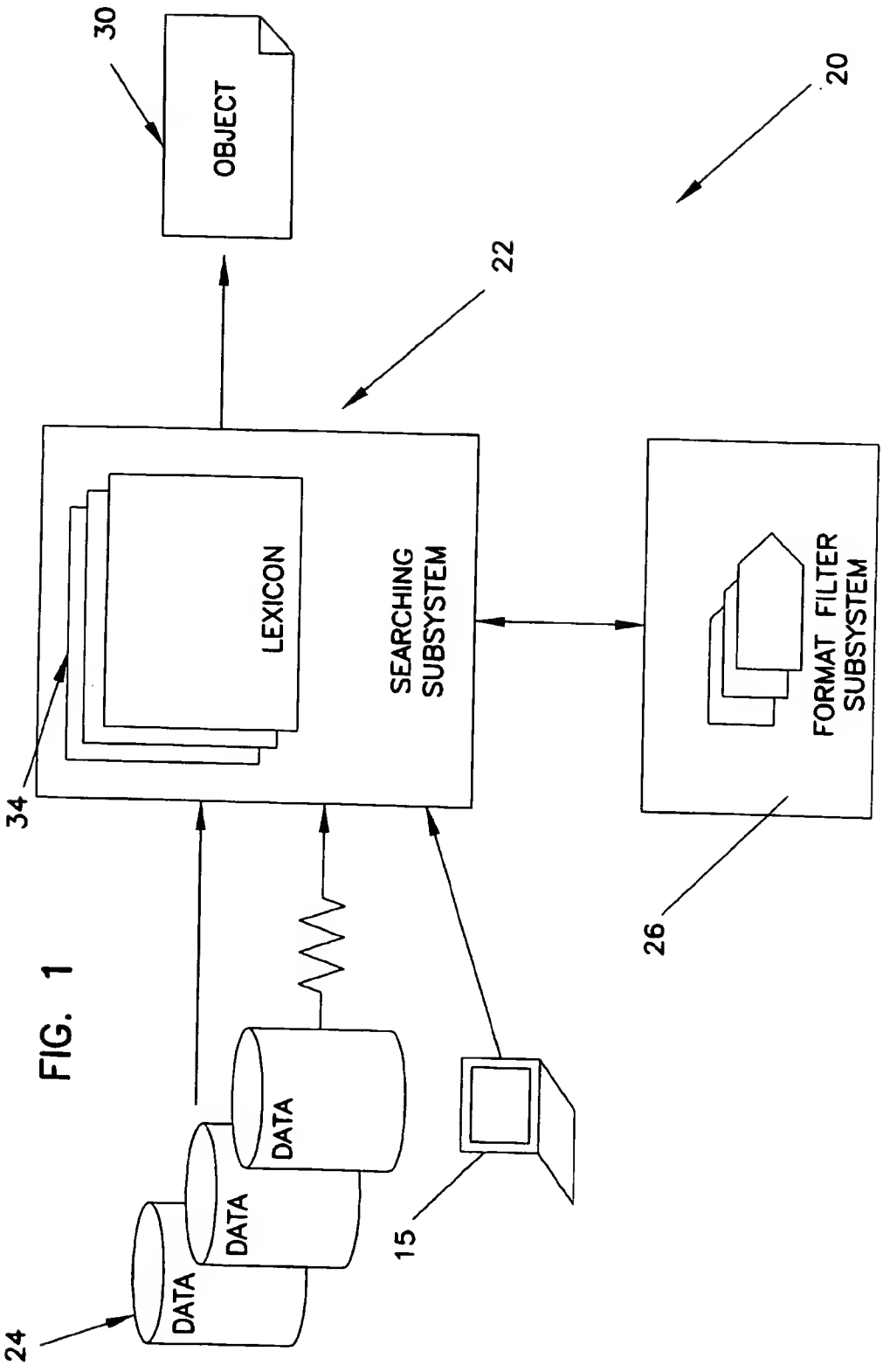

30
OBJECT
20
22
LEXICON
SEARCHING SUBSYSTEM
FORMAT FILTER SUBSYSTEM
34
26
DATA
DATA
DATA
24
15

FIG. 1

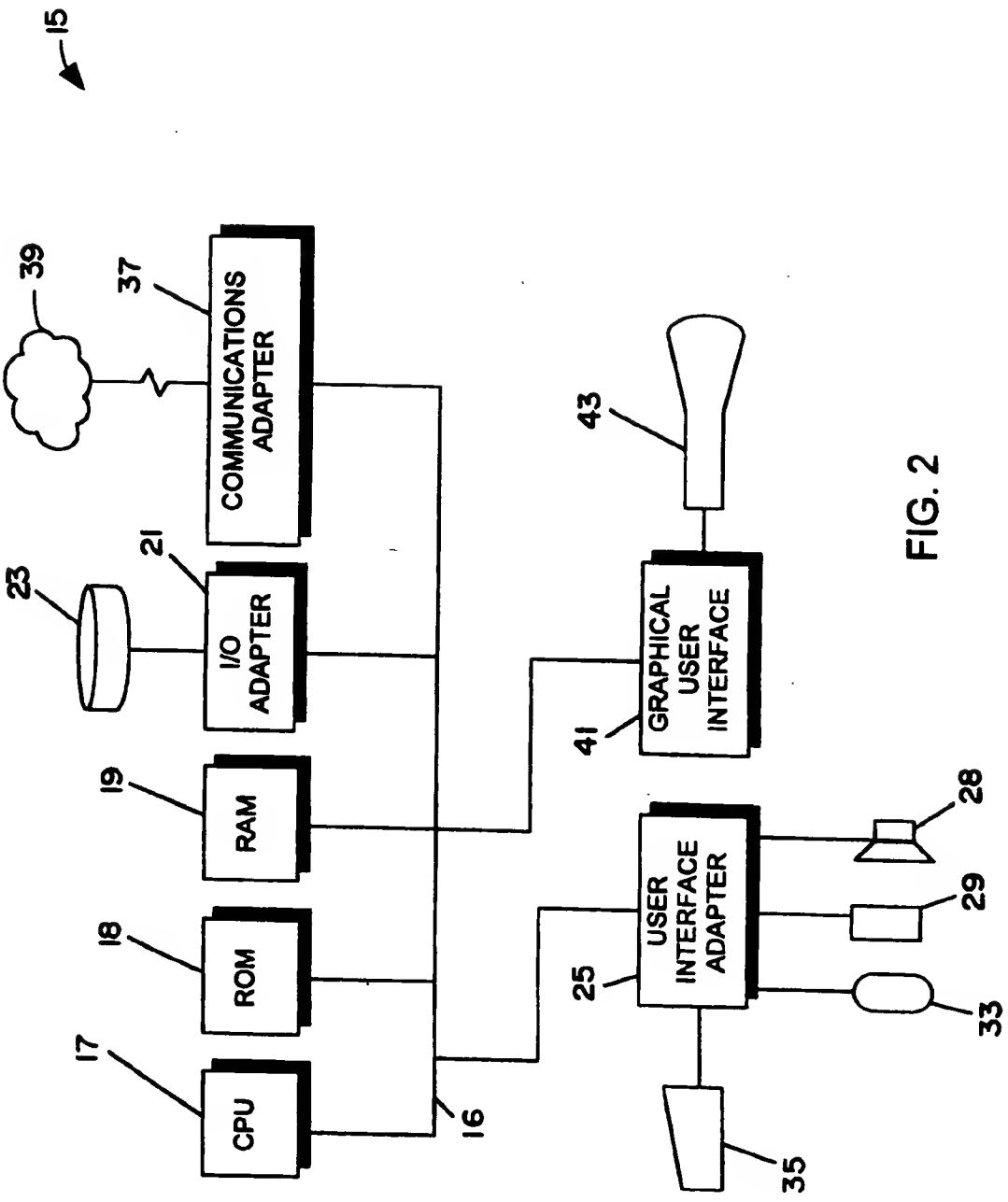

15
39
37
COMMUNICATIONS ADAPTER
43
21
23
I/O ADAPTER
GRAPHICAL USER INTERFACE
41
19
RAM
28
18
ROM
USER INTERFACE ADAPTER
29
25
17
CPU
16
33
35

FIG. 2

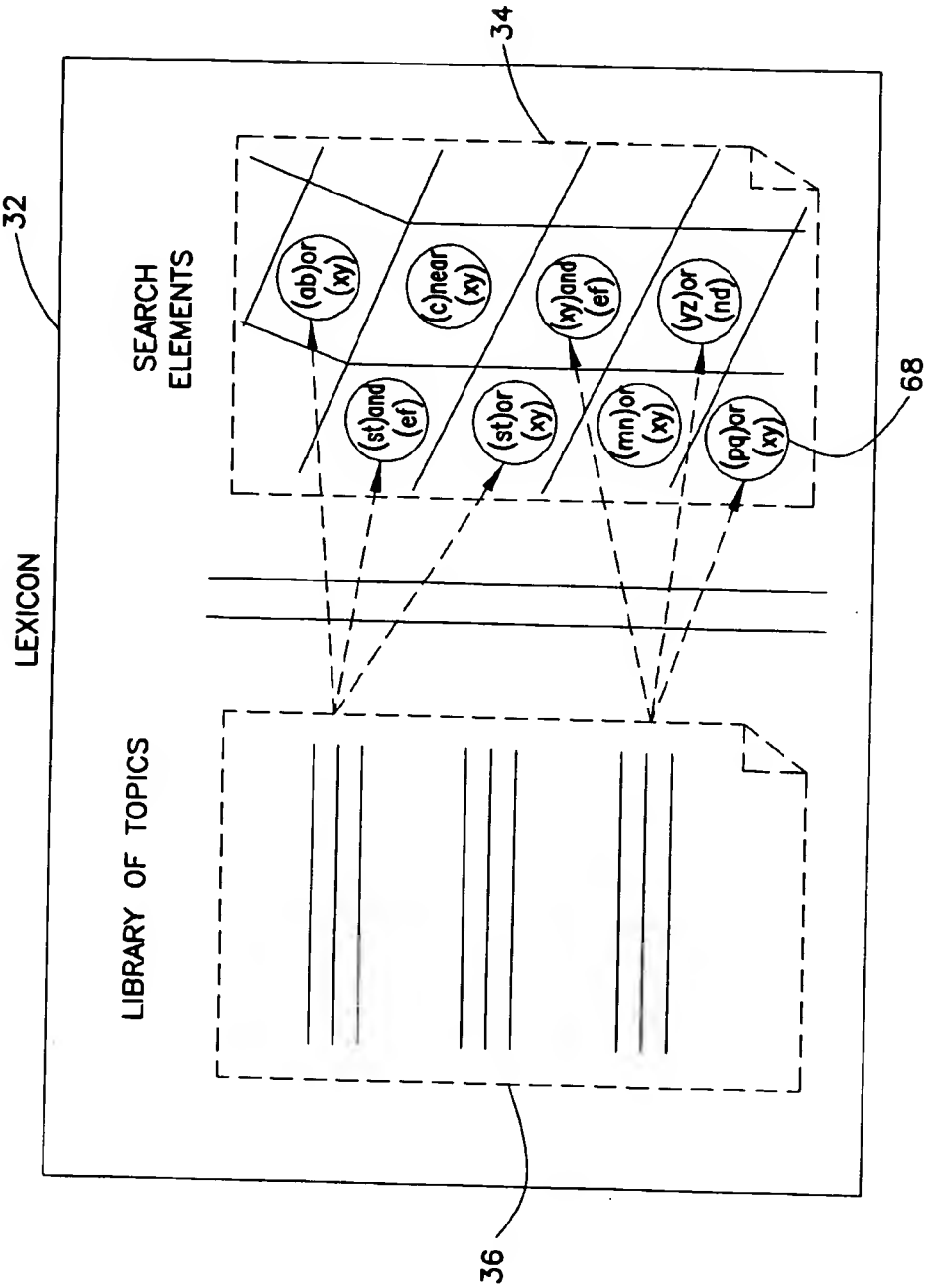

34
32
SEARCH ELEMENTS
(ab)or (xy)
(c)near (xy)
(xy)and (ef)
(yz)or (nd)
(st)and (ef)
(st)or (xy)
(mn)or (xy)
(pq)or (xy)
68
LEXICON
LIBRARY OF TOPICS
36

FIG. 3

FIG. 4

ENTERPRISE COMMUNITY

Operating Your Business–Budgeting

Topic	Subtopic	Logic Atom (expressed in Microsoft Indexing Server syntax)
Best Practices: Budgeting		((budgeting**) NEAR ("best practices" OR benchmarking** OR "best performers"))
Capital		((budgeting**) NEAR (capital OR asset** OR property OR resources OR facilities** OR "office space" OR furniture OR technology OR hardware OR software) NEAR (creating** OR developing** OR updating** OR "timing of updates" OR revising** OR revisions** OR "how to" OR "what about" OR contents** OR sections))
Facilities		((budgeting**) NEAR (facilities** OR building OR equipment OR plant OR structure OR "office space" OR furniture) NEAR (creating** OR developing** OR updating** OR "timing of updates" OR revising** OR revisions** OR "how to" OR "what about" OR contents** OR sections))
General Budgeting Techniques		(budgeting**) NEAR (techniques** OR methods** OR procedure** OR system** OR approach** OR methodology OR "time period"))
Operating		((budgeting**) NEAR (operating** OR people OR payroll) NEAR (creating** OR developing** OR updating** OR "timing of updates" OR revising** OR revisions** OR "how to" OR "what about" OR contents** OR sections))

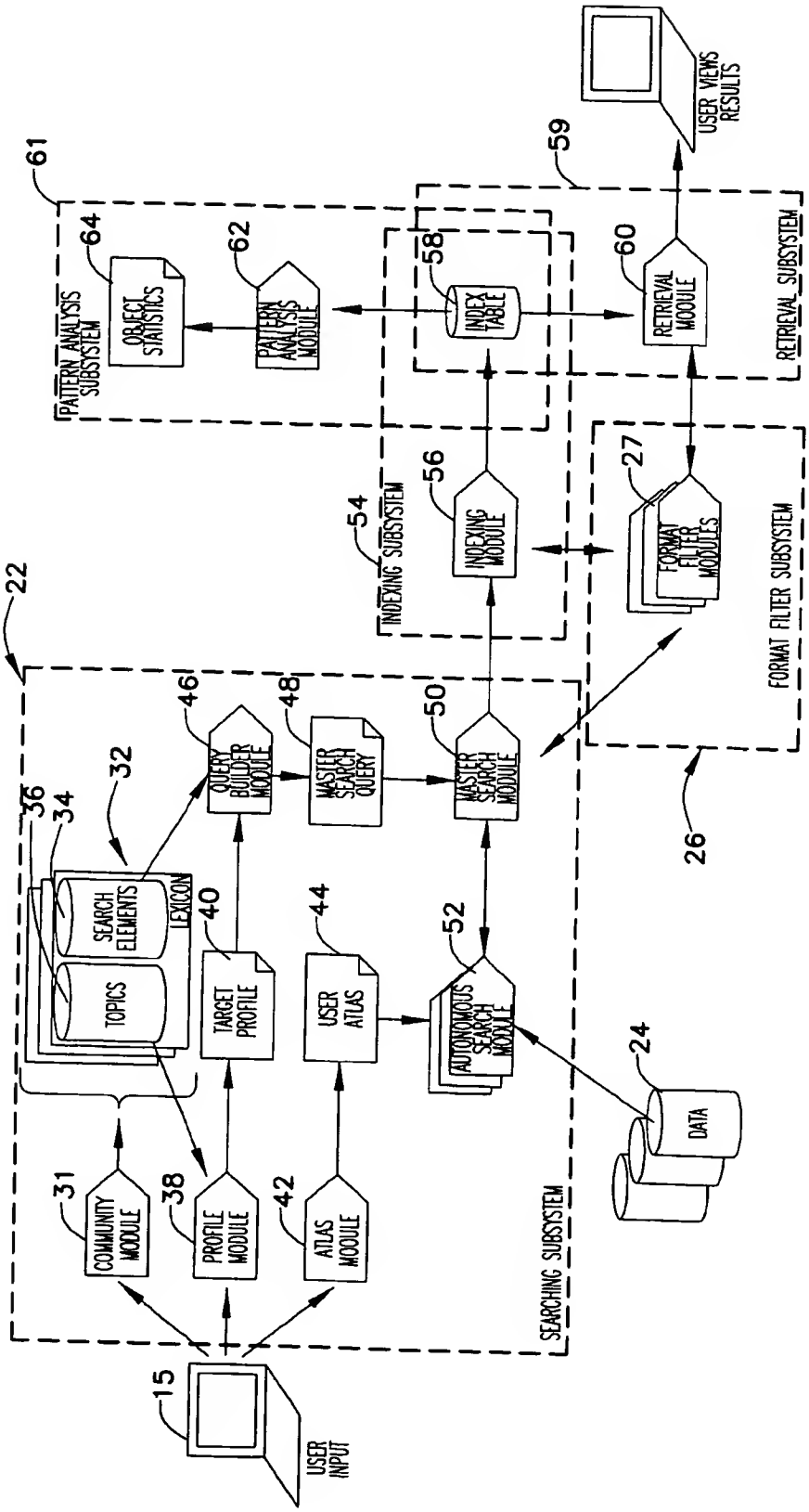

USER VIEWS RESULTS
59
61
64
62
58
60
PATTERN ANALYSIS SUBSYSTEM
OBJECT STATISTICS
PATTERN ANALYSIS MODULE
INDEX TABLE
RETRIEVAL MODULE
RETRIEVAL SUBSYSTEM
54
56
27
INDEXING SUBSYSTEM
INDEXING MODULE
FORMAT FILTER MODULES
FORMAT FILTER SUBSYSTEM
22
46
48
50
QUERY BUILDER MODULE
MASTER SEARCH QUERY
MASTER SEARCH MODULE
26
32
36
34
SEARCH ELEMENTS
LEXICON
TOPICS
40
44
52
TARGET PROFILE
USER ATLAS
AUTONOMOUS SEARCH MODULE
24
DATA
31
38
42
COMMUNITY MODULE
PROFILE MODULE
ATLAS MODULE
SEARCHING SUBSYSTEM
15
USER INPUT

FIG. 5

FIG. 6

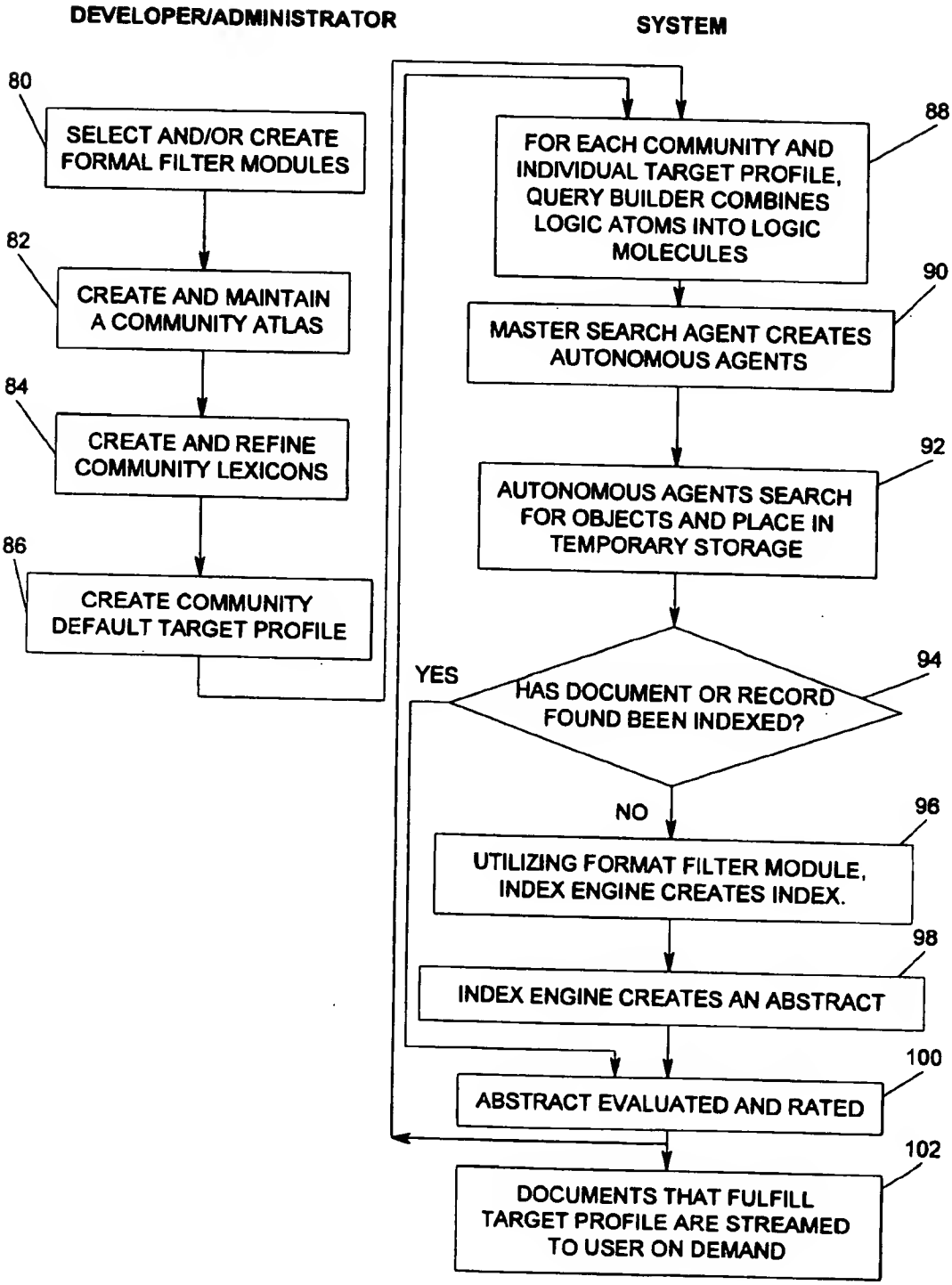

DEVELOPER/ADMINISTRATOR
SYSTEM
80
SELECT AND/OR CREATE FORMAL FILTER MODULES
82
CREATE AND MAINTAIN A COMMUNITY ATLAS
84
CREATE AND REFINE COMMUNITY LEXICONS
86
CREATE COMMUNITY DEFAULT TARGET PROFILE
88
FOR EACH COMMUNITY AND INDIVIDUAL TARGET PROFILE, QUERY BUILDER COMBINES LOGIC ATOMS INTO LOGIC MOLECULES
90
MASTER SEARCH AGENT CREATES AUTONOMOUS AGENTS
92
AUTONOMOUS AGENTS SEARCH FOR OBJECTS AND PLACE IN TEMPORARY STORAGE
94
HAS DOCUMENT OR RECORD FOUND BEEN INDEXED?
YES
NO
96
UTILIZING FORMAT FILTER MODULE, INDEX ENGINE CREATES INDEX.
98
INDEX ENGINE CREATES AN ABSTRACT
100
ABSTRACT EVALUATED AND RATED
102
DOCUMENTS THAT FULFILL TARGET PROFILE ARE STREAMED TO USER ON DEMAND

FIG. 7

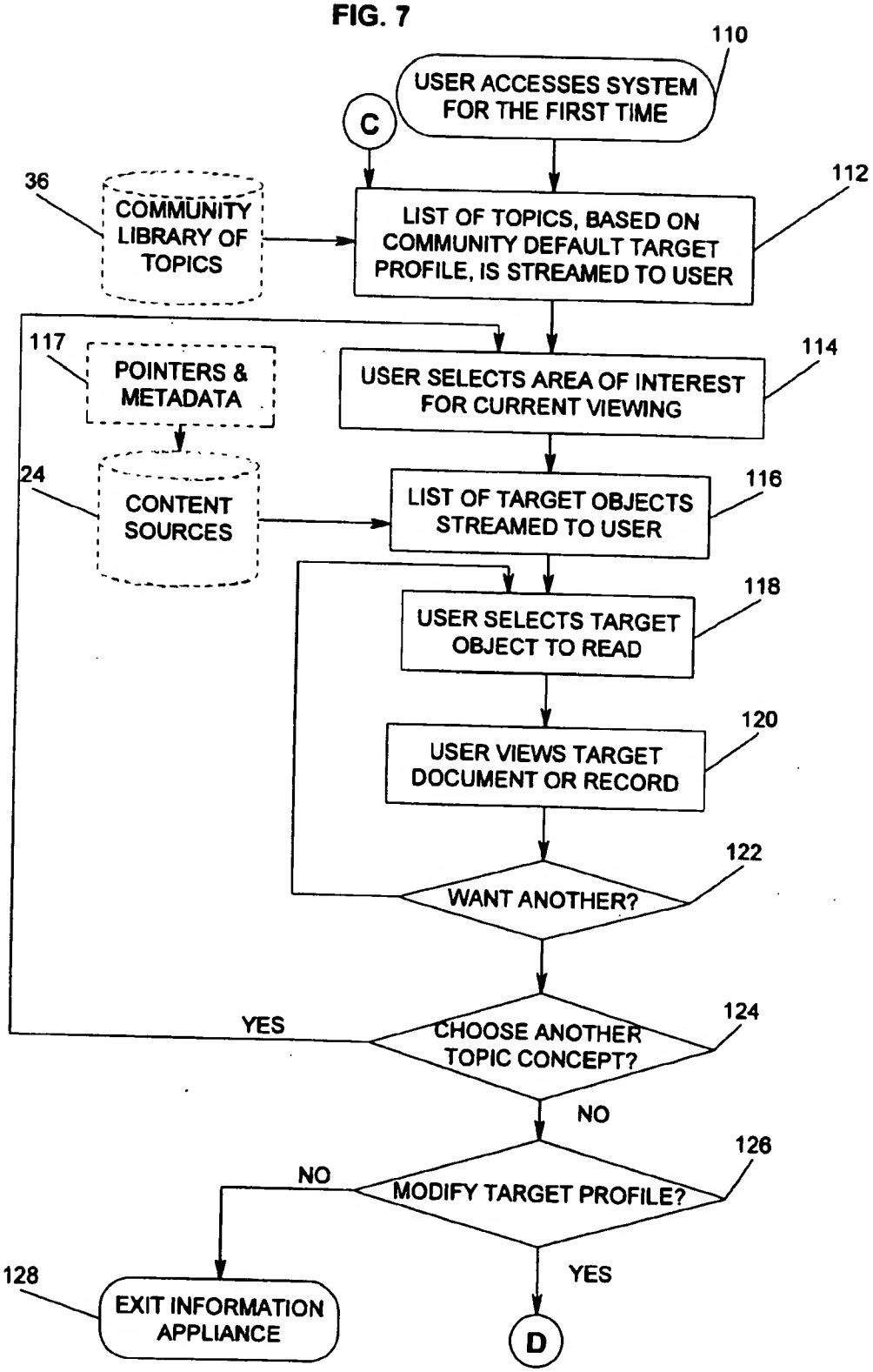

110
USER ACCESSES SYSTEM FOR THE FIRST TIME
C
36
COMMUNITY LIBRARY OF TOPICS
112
LIST OF TOPICS, BASED ON COMMUNITY DEFAULT TARGET PROFILE, IS STREAMED TO USER
117
POINTERS & METADATA
114
USER SELECTS AREA OF INTEREST FOR CURRENT VIEWING
24
CONTENT SOURCES
116
LIST OF TARGET OBJECTS STREAMED TO USER
118
USER SELECTS TARGET OBJECT TO READ
120
USER VIEWS TARGET DOCUMENT OR RECORD
122
WANT ANOTHER?
YES
124
CHOOSE ANOTHER TOPIC CONCEPT?
NO
126
NO
MODIFY TARGET PROFILE?
YES
128
EXIT INFORMATION APPLIANCE
D

FIG. 8

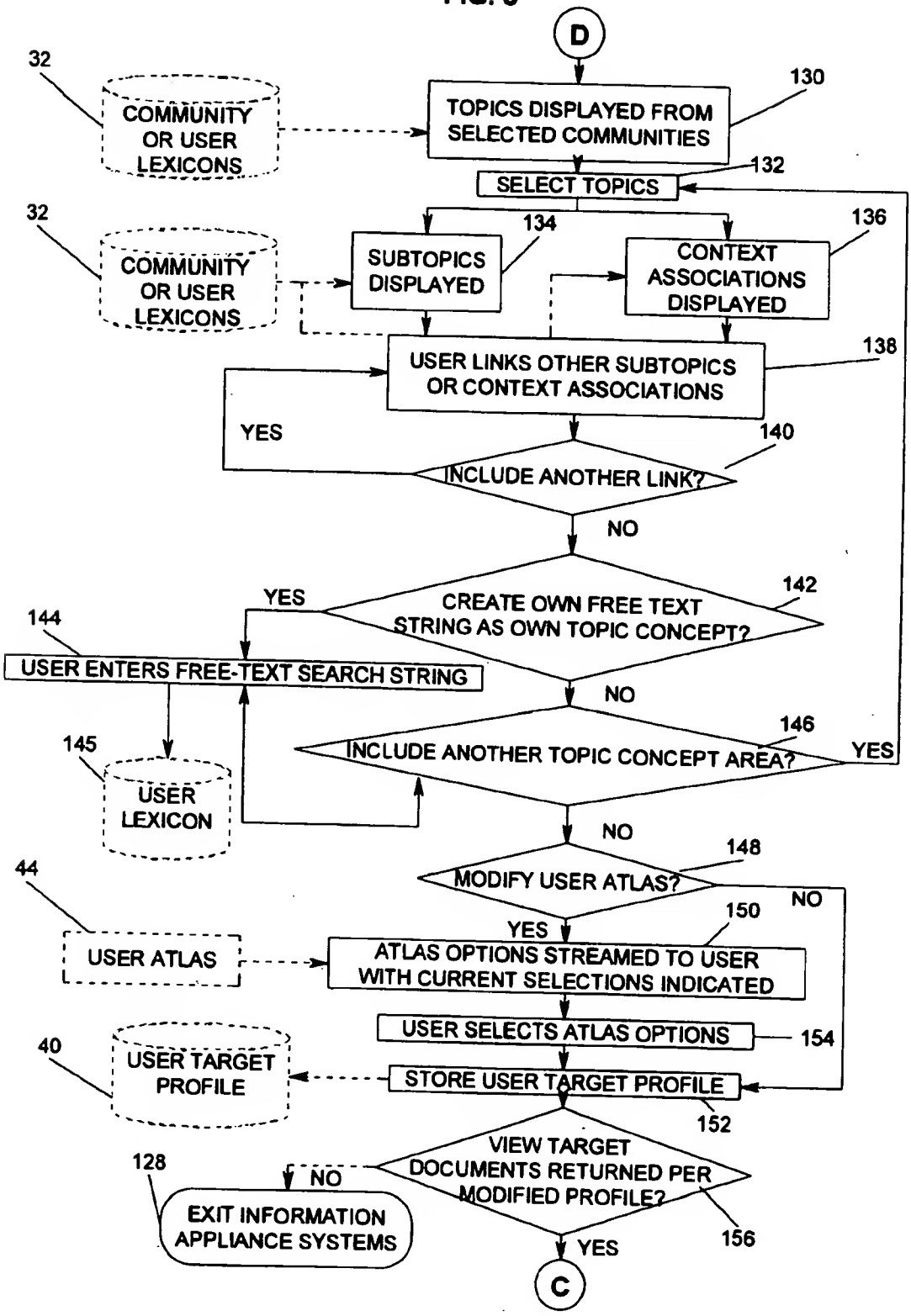

D
32
COMMUNITY OR USER LEXICONS
TOPICS DISPLAYED FROM SELECTED COMMUNITIES
130
SELECT TOPICS
132
32
COMMUNITY OR USER LEXICONS
SUBTOPICS DISPLAYED
134
CONTEXT ASSOCIATIONS DISPLAYED
136
USER LINKS OTHER SUBTOPICS OR CONTEXT ASSOCIATIONS
138
YES
INCLUDE ANOTHER LINK?
140
NO
YES
CREATE OWN FREE TEXT STRING AS OWN TOPIC CONCEPT?
142
144
USER ENTERS FREE-TEXT SEARCH STRING
NO
INCLUDE ANOTHER TOPIC CONCEPT AREA?
146
YES
145
USER LEXICON
NO
MODIFY USER ATLAS?
148
NO
44
YES
150
USER ATLAS
ATLAS OPTIONS STREAMED TO USER WITH CURRENT SELECTIONS INDICATED
USER SELECTS ATLAS OPTIONS
154
40
USER TARGET PROFILE
STORE USER TARGET PROFILE
152
VIEW TARGET DOCUMENTS RETURNED PER MODIFIED PROFILE?
128
NO
156
EXIT INFORMATION APPLIANCE SYSTEMS
YES
C

FIG. 9

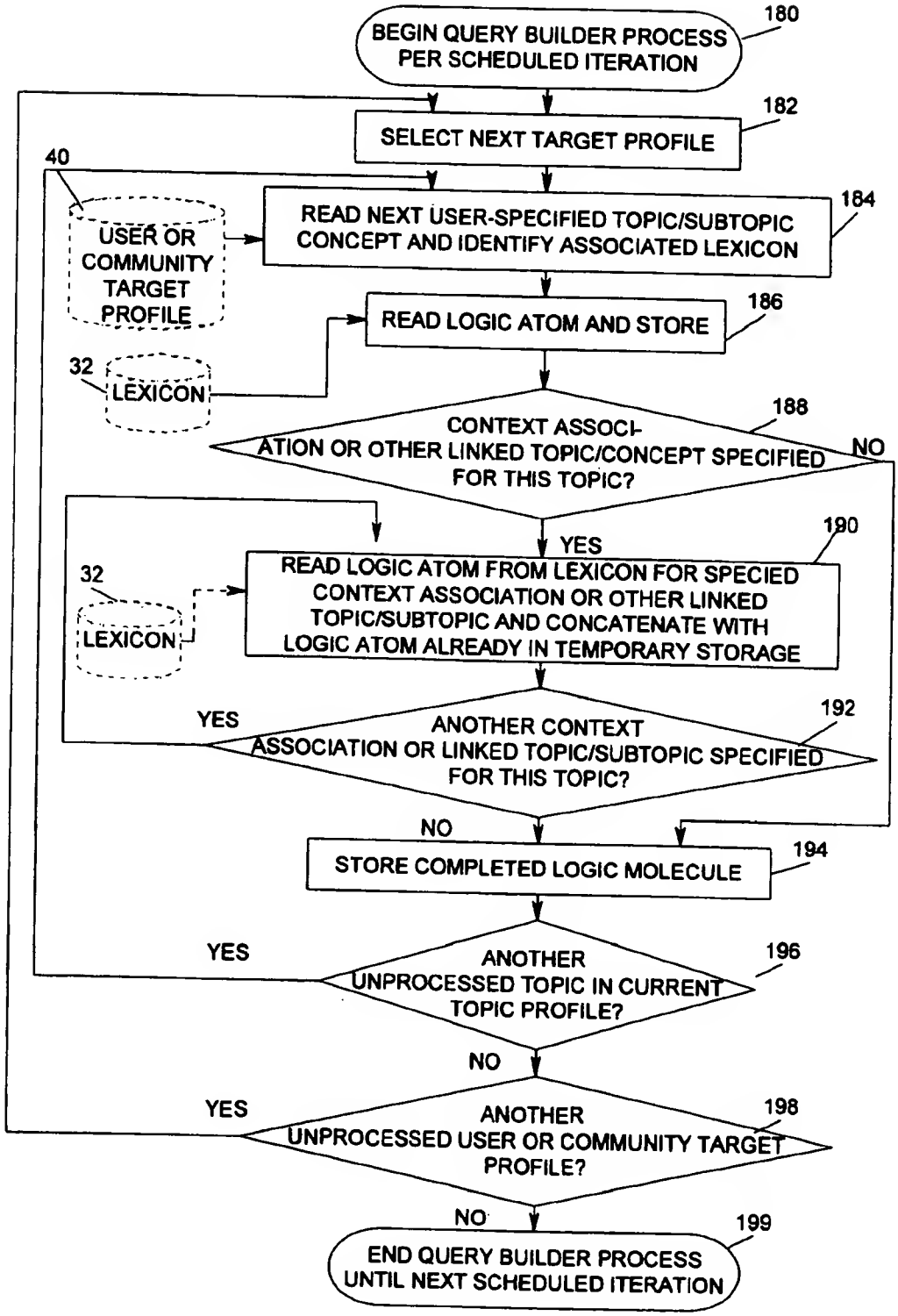

BEGIN QUERY BUILDER PROCESS PER SCHEDULED ITERATION
180
SELECT NEXT TARGET PROFILE
182
40
USER OR COMMUNITY TARGET PROFILE
READ NEXT USER-SPECIFIED TOPIC/SUBTOPIC CONCEPT AND IDENTIFY ASSOCIATED LEXICON
184
READ LOGIC ATOM AND STORE
186
32
LEXICON
CONTEXT ASSOCIATION OR OTHER LINKED TOPIC/CONCEPT SPECIFIED FOR THIS TOPIC?
188
NO
YES
READ LOGIC ATOM FROM LEXICON FOR SPECIED CONTEXT ASSOCIATION OR OTHER LINKED TOPIC/SUBTOPIC AND CONCATENATE WITH LOGIC ATOM ALREADY IN TEMPORARY STORAGE
190
32
LEXICON
YES
ANOTHER CONTEXT ASSOCIATION OR LINKED TOPIC/SUBTOPIC SPECIFIED FOR THIS TOPIC?
192
NO
STORE COMPLETED LOGIC MOLECULE
194
YES
ANOTHER UNPROCESSED TOPIC IN CURRENT TOPIC PROFILE?
196
NO
YES
ANOTHER UNPROCESSED USER OR COMMUNITY TARGET PROFILE?
198
NO
END QUERY BUILDER PROCESS UNTIL NEXT SCHEDULED ITERATION
199

FIG. 10

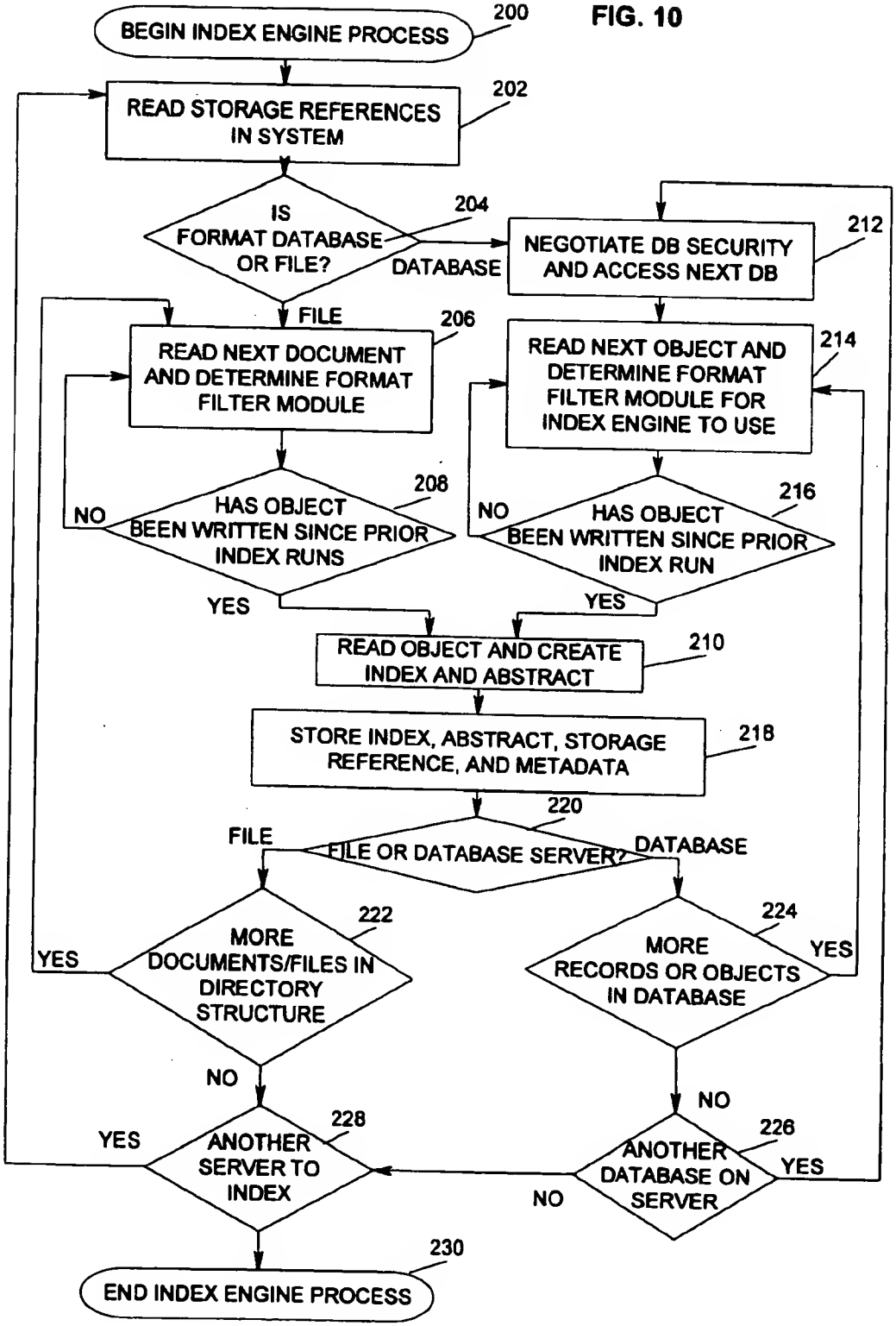

BEGIN INDEX ENGINE PROCESS
200
READ STORAGE REFERENCES IN SYSTEM
202
IS FORMAT DATABASE OR FILE?
204
DATABASE
FILE
NEGOTIATE DB SECURITY AND ACCESS NEXT DB
212
READ NEXT DOCUMENT AND DETERMINE FORMAT FILTER MODULE
206
READ NEXT OBJECT AND DETERMINE FORMAT FILTER MODULE FOR INDEX ENGINE TO USE
214
HAS OBJECT BEEN WRITTEN SINCE PRIOR INDEX RUNS
208
NO
YES
HAS OBJECT BEEN WRITTEN SINCE PRIOR INDEX RUN
216
NO
YES
READ OBJECT AND CREATE INDEX AND ABSTRACT
210
STORE INDEX, ABSTRACT, STORAGE REFERENCE, AND METADATA
218
FILE OR DATABASE SERVER?
220
FILE
DATABASE
MORE DOCUMENTS/FILES IN DIRECTORY STRUCTURE
222
YES
NO
MORE RECORDS OR OBJECTS IN DATABASE
224
YES
NO
ANOTHER SERVER TO INDEX
228
YES
ANOTHER DATABASE ON SERVER
226
NO
YES
END INDEX ENGINE PROCESS
230

FIG. 11

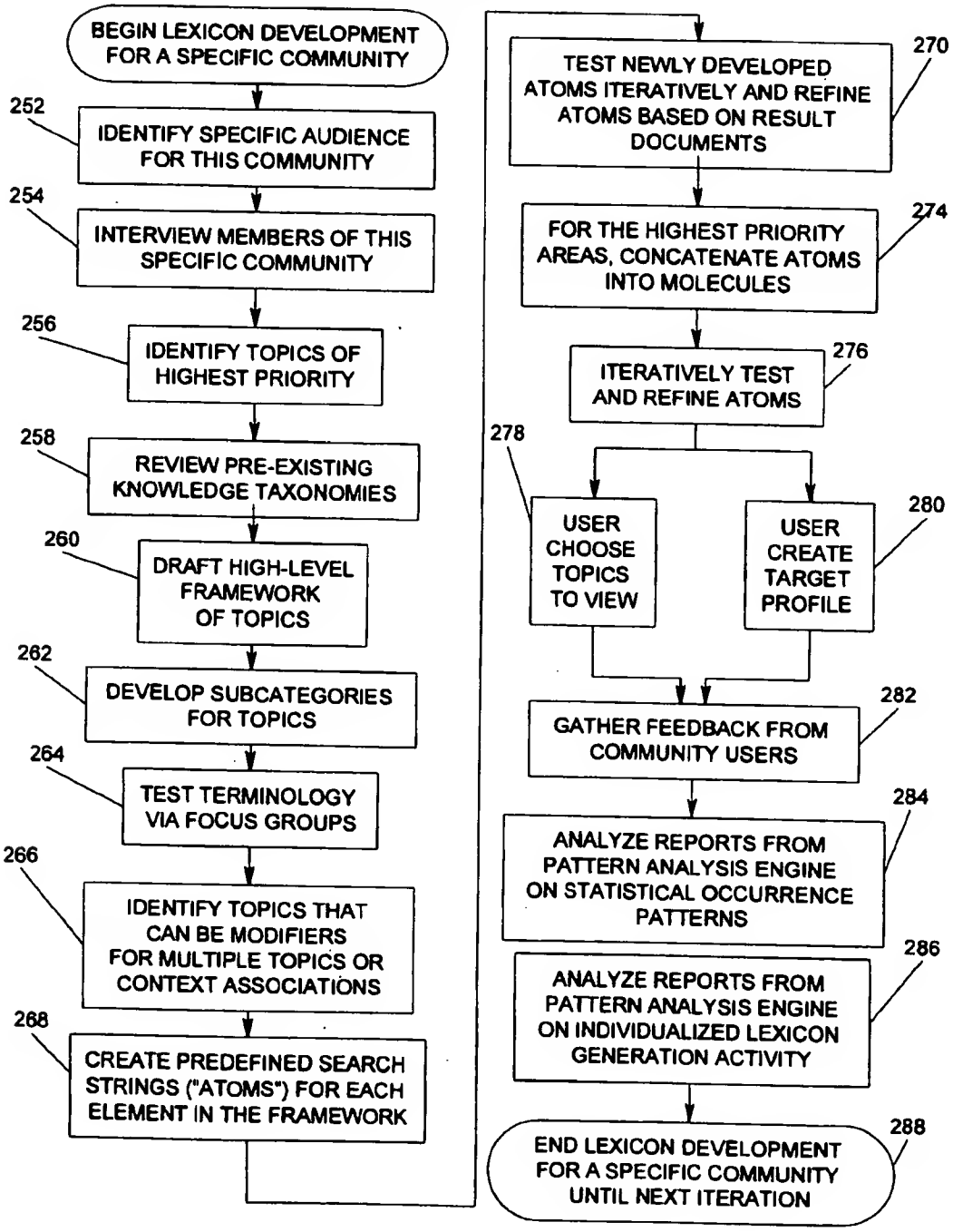
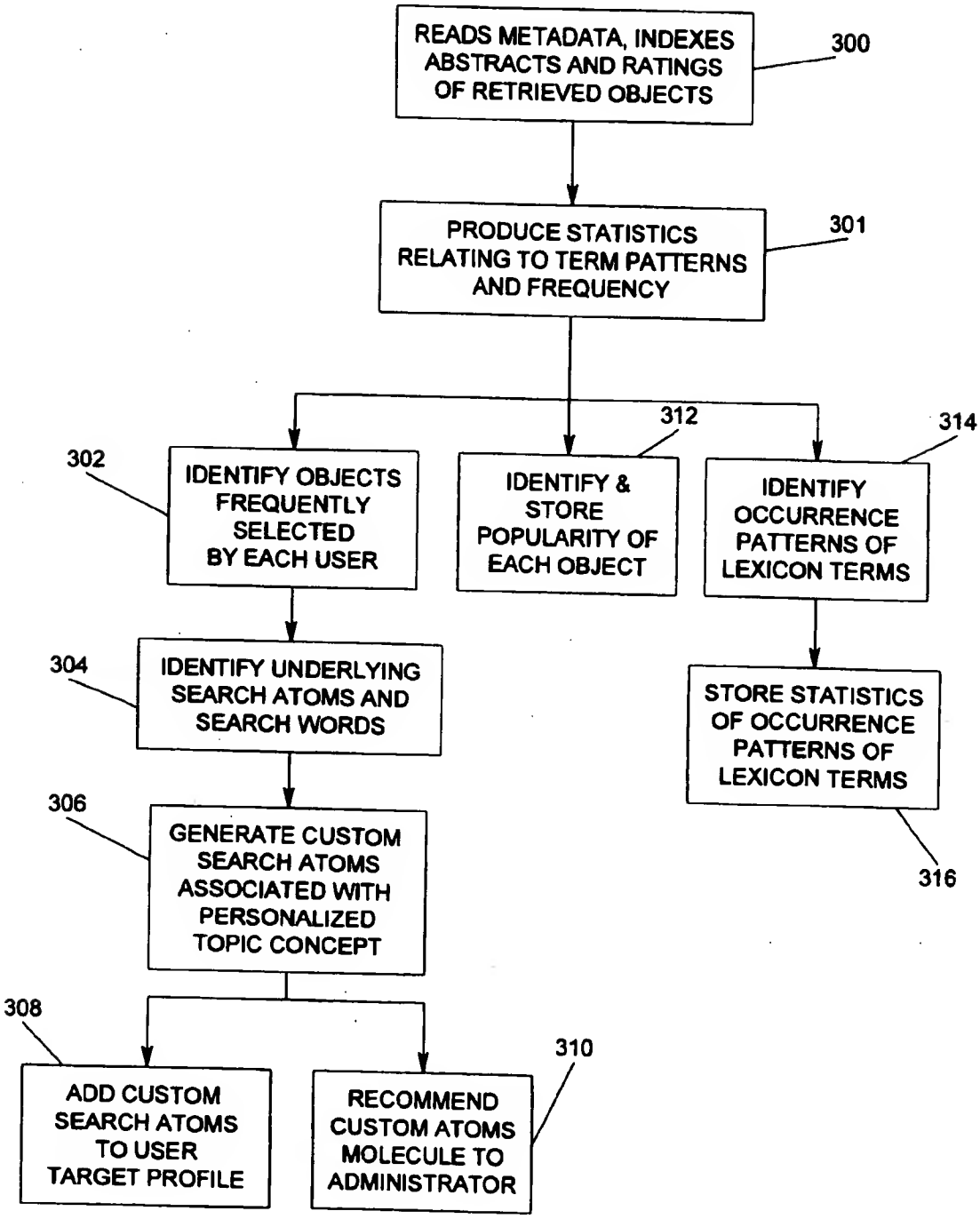

BEGIN LEXICON DEVELOPMENT FOR A SPECIFIC COMMUNITY
252
IDENTIFY SPECIFIC AUDIENCE FOR THIS COMMUNITY
254
INTERVIEW MEMBERS OF THIS SPECIFIC COMMUNITY
256
IDENTIFY TOPICS OF HIGHEST PRIORITY
258
REVIEW PRE-EXISTING KNOWLEDGE TAXONOMIES
260
DRAFT HIGH-LEVEL FRAMEWORK OF TOPICS
262
DEVELOP SUBCATEGORIES FOR TOPICS
264
TEST TERMINOLOGY VIA FOCUS GROUPS
266
IDENTIFY TOPICS THAT CAN BE MODIFIERS FOR MULTIPLE TOPICS OR CONTEXT ASSOCIATIONS
268
CREATE PREDEFINED SEARCH STRINGS ("ATOMS") FOR EACH ELEMENT IN THE FRAMEWORK
270
TEST NEWLY DEVELOPED ATOMS ITERATIVELY AND REFINE ATOMS BASED ON RESULT DOCUMENTS
274
FOR THE HIGHEST PRIORITY AREAS, CONCATENATE ATOMS INTO MOLECULES
276
ITERATIVELY TEST AND REFINE ATOMS
278
USER CHOOSE TOPICS TO VIEW
280
USER CREATE TARGET PROFILE
282
GATHER FEEDBACK FROM COMMUNITY USERS
284
ANALYZE REPORTS FROM PATTERN ANALYSIS ENGINE ON STATISTICAL OCCURRENCE PATTERNS
286
ANALYZE REPORTS FROM PATTERN ANALYSIS ENGINE ON INDIVIDUALIZED LEXICON GENERATION ACTIVITY
288
END LEXICON DEVELOPMENT FOR A SPECIFIC COMMUNITY UNTIL NEXT ITERATION

FIG. 12


READS METADATA, INDEXES ABSTRACTS AND RATINGS OF RETRIEVED OBJECTS
300
PRODUCE STATISTICS RELATING TO TERM PATTERNS AND FREQUENCY
301
302
IDENTIFY OBJECTS FREQUENTLY SELECTED BY EACH USER
312
IDENTIFY & STORE POPULARITY OF EACH OBJECT
314
IDENTIFY OCCURRENCE PATTERNS OF LEXICON TERMS
304
IDENTIFY UNDERLYING SEARCH ATOMS AND SEARCH WORDS
STORE STATISTICS OF OCCURRENCE PATTERNS OF LEXICON TERMS
316
306
GENERATE CUSTOM SEARCH ATOMS ASSOCIATED WITH PERSONALIZED TOPIC CONCEPT
308
310
ADD CUSTOM SEARCH ATOMS TO USER TARGET PROFILE
RECOMMEND CUSTOM ATOMS MOLECULE TO ADMINISTRATOR

SEARCH AND RETRIEVAL INFORMATION SYSTEM AND METHOD

FIELD OF THE INVENTION

The present invention relates generally to information systems, and more particularly to a broad-based information system for searching and automatically retrieving information stored across multiple platforms while parsing and filtering that information according to a particular community of interest.

BACKGROUND OF THE INVENTION

Easy and efficient access to information has become essential to maintaining an effective organization. Most information is stored and accessed from discrete sources on internal networks where the organization's financial and document systems are maintained. However, because of increased use of the Internet and newsfeeds, the demand for external information has also grown. To access these different sources of information, a wide variety of search and retrieval systems are used. However, many systems often fail to deliver information which is relevant to the specific needs of each user. Furthermore, many systems are limited by their inability to traverse different platforms and operating systems to search multiple sources of information and deliver the information to a single location. These limitations are counterproductive to the user's needs for an easy and efficient method of accessing information across multiple platforms.

Many existing search and retrieval systems require the user to specify a query statement on search criteria. Typically, such systems enhance the user supplied query using word associations similar to a thesaurus. However, because these word associations are generic, these systems often do not focus the search to the specific needs of the user. Consequently, for the search to yield germane information, the user typically either sifts through the information to determine its relevancy or performs several iterations of the search each time refining the search strings. This process is time-consuming and inefficient.

Similarly, some systems allow users to personalize their search and achieve a high degree of specificity. However, the user must learn and use complex search syntax that is often difficult for the user to understand, search iteratively, and carefully craft the query in order to obtain specific results. For example, many search systems require the user to input search strings using Boolean operators. Hence, for the search to be effective, the user must be proficient with the usage of Boolean operators. Otherwise, the search may not produce useful information and may be too time consuming.

Another problem with many current search systems is that they require individual user-initiated queries, instead of providing a flow of highly relevant information on numerous topics. Separate queries are usually required for separate topics, and these queries need to be repeated by the user at appropriate time intervals. This need to initiate a separate query for each topic of interest further lengthens the process and exacerbates inefficiency.

Furthermore, many existing search and retrieval systems are limited to searching certain sources of information. This severely confines the usefulness of these systems because users are often required to perform the same searches on different systems to access all potential sources of information, both internal to and external to the user's system. The inefficiencies inherent in this process are compounded in light of the inability of most systems to retrieve

information relevant to the needs of the user, and the need for the user to initiate separate queries for each topic of interest.

No system currently exists that retrieves a flow of information from sources originating from multiple platforms and operating systems while ensuring its relevancy to the user. While users currently have access to many sources of information for managing their operations, the sources of information are varied. Organizations need access to a wider range of information and an ability to tailor that information to the specific needs of the user. An information system is needed wherein all potential sources of information can be easily and automatically searched, and only relevant information is retrieved and displayed to the user.

The present invention provides a system for search and retrieval of electronic objects, the objects including electronically encoded information. The system is made up of at least a searching subsystem, which includes one or more electronic lexicons in a memory within the system, and a format filter subsystem coupled to the searching subsystem. The electronic lexicon provides predefined search query elements that are contextualized for specific communities, to identify objects that are relevant to the selections of specific individuals. The format filter subsystem includes several format filter modules that operate to identify a format of an electronic object and then select a format filter module that will enable the system to search the object using the search logic elements within the lexicon.

The present invention also provides for a method for search and retrieval of electronic objects, including identifying a format of an object to be searched, selecting a format filter module that is configured to enable searching, and searching the object using predefined search elements that are found in an electronic lexicon. An aspect of another embodiment of the method is that retrieved objects may be delivered to the user in a single viewing format.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention may be more completely understood in consideration of the detailed description of various embodiments of the invention which follows in connection with the accompanying drawings, in which:

FIG. **1** is an overview block diagram of a search and retrieval information system;

FIG. **2** is a block diagram of a personal computer system;

FIG. **3** is a functional drawing of a lexicon using a search and retrieval information system;

FIG. **4** shows an example of several lexicon entries in a search and retrieval information system.

FIG. **5** is a more detailed block diagram of a search and retrieval information system;

FIG. **6** is a flowchart of a search and retrieval information method;

FIGS. **7** and **8** are flowcharts of a process by which users personalize a target profile in a search and retrieval information system and method;

FIG. **9** is a flowchart of steps taken by a query builder module in a search and retrieval information system and method;

FIG. **10** is a flowchart showing steps taken by an indexing module in a search and retrieval information system and method;

FIG. **11** shows steps taken by an administrator to develop a lexicon in a search and retrieval information system and method;

FIG. 12 is a flowchart showing steps taken by a pattern analysis module in a search and retrieval information system and method;

While the present invention is amenable to various modifications and alternative forms, specifics thereof have been shown by way of example in the drawings and will be described in detail. It should be understood, however, that the intention is not to limit the invention to the particular embodiments described. On the contrary, the invention is to cover all modifications, equivalents, and alternatives falling within the spirit and scope of the invention as defined by the appended claims.

DETAILED DESCRIPTION OF THE VARIOUS EMBODIMENTS

The invention is believed to be applicable to a variety of systems and arrangements which search and automatically retrieve information. The invention has been found to be particularly advantageous in application environments where system users require access to information which exists on different platforms and operating systems. While the present invention is not so limited, an appreciation of various aspects of the invention is best gained through a discussion of various application examples operating in such an environment.

Information and knowledge are essential raw materials and assets among many of today's workers, executives, and business owners. Current information management systems are typically focused on only one type of information repository. These management systems require that data be categorized or formatted in a specific way at the time of storage in order for that information to be available for search and retrieval. Therefore, current information management systems are of limited utility to most users, who need to obtain useful information from multiple types of sources.

Many executives and business owners are unable to obtain the desired kind or quantity of relevant information needed to serve business clients, help and improve their businesses, and to manage the activities of employee teams. To serve clients, many executives are in need of highly specific business and industry news and information. To help them run their own businesses, they need timely activity and status reports from their employee teams, which requires the ability to search on their own internal databases. A wide variety of financial reporting information is also essential to many executives' and business owners' success.

It is also important that information seekers are able to easily access information using a personal computer. Another complaint about current information management systems is that they are difficult to connect to and difficult to direct toward highly specific information needs. Many information management systems require that a search query be written in complex Boolean logic statements. For highly specific information requirements, very complex Boolean statements and repeated alternate search strings are often currently required. Even where a search engine may incorporate natural language capabilities, users may need repeated alternative queries and iterative refinement of the search query so that it is specific to a given industry, product type, geographic location, or time period.

The present information search and retrieval system and method is designed to address these shortcomings in current information management systems. The present information search and retrieval system and method will also be referred to as an information appliance. The term information appliance is intended to refer to the system as a whole. The information appliance described can search diverse types of data and files because it is provided with format filter modules in order to be able to access information in various formats. For example, the information appliance may easily search documents that are internal to the user's system, or external commercial databases. Format filter modules may first identify an object's format, then use a specific filter module to read and search the object. The term "object," or "electronic object," will be used to refer to any type of electronic information that can be searched and accessed. Examples of electronic objects could be text documents such as newspaper articles, trade journal articles, report documents, or financial reporting information within an electronic database. In one embodiment the information appliance may also access, for example, Domino Notes® documents, relational database tables, object-oriented records, and other documents, records and databases.

In order to help a user obtain highly relevant information, without knowledge of complex Boolean search string construction, the information appliance provides predefined search elements designed to identify electronic objects that will be most useful to the user's community. A list of topics for each community allows the user to easily describe the desired range of the search. Each topic may be associated with the predefined search criteria that is highly specific to the user's industry or community. An example of a topic is "general budgeting techniques." Some topics are more broadly indicative of an industry context, such as a geographical context like "South America" or a time frame like "Spring, 1996." Some topics are more like subtopics, such as the subtopic "pricing strategy" being more specific than the topic "marketing plan."

Each topic typically is linked to a predefined search query that is designed to gather information on the topic relevant to the user. The search query may contain words frequently found within a discussion relating to the topic and may look for those words in the object being searched. When the user specifies two topics, the information appliance may link the two search queries associated with the two topics and thereby execute a highly specific search. Throughout this description, the term "topic" will be used to refer to not only a topic but a subtopic and context association, where each is linked to a stored search element. A topic could also be a type of document, such as receivables reports or status reports. If the topic is a type of report, the search query may look for certain types of document names, or indicative words within the document. A topic may also be a specific document, storage location, or address, such as a web site address, where relevant information resides. In this case, the search element may be just the document name and location, so that the document is retrieved when the search is executed. A user creates a new topic when the user combines a topic with a context association. A user may also create a new topic by defining a new search query entirely. The search query associated with each topic is herein termed a search atom or search element. When several search atoms are linked, the resulting highly specific search string is termed a search molecule.

FIGS. 3 and 4 illustrate a lexicon. FIG. 3 shows a lexicon 32 as being made up of a library of topics 36 and a set of search elements 34. Within the present system, a lexicon is typically a storage framework of search elements or search atoms, each linked to one or more topics in the library of topics. When a topic is chosen by the user to be searched, the search query that is linked to the topic is used to carry out the search. When a topic appears to the user in the interface

for viewing selection, the topic may be linked to a single search atom, or the topic may be linked to a compounded query or search molecule which incorporates more than one independent search query. The type of links associated with each topic depends on the topic.

FIG. **4** shows an example of the topics being linked to the search elements in a different way of showing lexicon **32**. The community lexicon shown in FIG. **4** is for a community termed Enterprise. Some topics within the Enterprise community lexicon include best practices: budgeting, capital, facilities, and operating. It is also possible to have more specific subtopics within each topic. The right-hand side of the chart of FIG. **4** shows search queries that are linked to each topic. For example, the search query linked to the topic "capital" is very lengthy and lists examples of capital that might be found in a discussion of capital in the Enterprise context. In one embodiment of the present information appliance, the user will be able to designate the topic "capital" within the community "Enterprise" and have the benefit of the complex preprogrammed search criteria.

FIG. **1** illustrates one particular embodiment of an information system for searching and retrieving electronic objects across multiple platforms and operating systems. Referring to FIG. **1**, the system **20** shown includes a searching subsystem **22** which is capable of accessing data **24** stored across multiple platforms and operating systems. Some data accessed may be internal to the system, while some data may be accessible through a remote communication line. Because the system accesses data resources from various platforms, the searching subsystem interfaces with a format filter subsystem **26** to search data of varying formats. Electronic lexicon **34** is used by the searching subsystem to search the data and identify electronic objects **30** which are specifically relevant to the information needs of the user. Thus, the lexicon typically stores search elements that reflect the user's community of interests and returns only those electronic objects which are specific to the user's interest and the interest of the user's community.

The information appliance may be used with many different computer systems. In a client/server system, each user is provided with a user terminal, such as a personal computer, which may be linked to a modem, communication lines, network lines, a central processor, and databases. An NT or UNIX server, for example, may be used with this system. The user terminal provides the user with a way to input the user's preferences to the information appliance and a way to view the electronic objects retrieved. The preferred embodiment of the information appliance may be practiced with the user terminal **15** of FIG. **1** being a personal computer such as an IBM®, Compaq®, Dell®, or Apple® Macintosh® personal computer. As previously indicated, user terminal **15** may preferably be part of a client/server system. A representative hardware environment of the user terminal is shown in FIG. **2**. The preferred hardware configuration includes a central processing unit **17**, such as a microprocessor, and a number of other units interconnected by, for example, a system bus **16**. The computer of a terminal **15** may also he spread out over one or more interconnected computers or computer systems.

The user terminal shown in FIG. **2** also includes a Random Access Memory in (RAM) **19**, Read Only Memory (ROM) **18**, and an I/O adapter **21** for connecting peripheral devices such as disk storage units **23** to the bus **16**. A user interface adapter **25** for connecting several input devices is also included. Examples of possible input devices connected to the user interface adapter **25** include a keyboard **35**, a mouse **29**, a speaker **28**, a microphone **33**, and/or other user interface devices such as a touch screen or voice interface (not shown). A communication adapter **37** is included for connecting the user terminal to a communication network link **39**. A graphical user interface **41** is also connected to the system bus **16** and provides the connection to a display device **43**. It will be apparent to those in the art that the mouse **29** may be a typical mouse as known in the industry, a trackball, light pen, or the like.

The user terminal typically has resident thereon an operating system such as Windows®, Windows NT®, Apple System 7®, IBM OS/2®, or UNIX® software. The network also has a resident operating system, for example, Novell® Netware or Novell® Intranetware, among other possibilities. In the preferred environment, the desktop typically has Internet browser software, such as MS Internet Explorer or Netscape Navigator. In the alternative, the network software operating system may not be available separate from the work station operating system, and the network operating system may have an integrated Internet browser. Other alternatives for client and server software include Oracle® or Microsoft Sequel Server.

A networked personal computer environment, a client/server system, a mainframe terminal environment, WEB TV terminal environment, dumb terminal environments, a networked computer environment that is connected to an Internet site or a personal computer alone could be used to implement the information appliance. Any type of system that allows the user to receive target objects and documents and use an input device to set up a user profile could be used with this system. Depending upon the user's needs, a client/server system may be the most preferable computer system for implementing the information appliance.

FIG. **5** shows a more detailed diagram of several subsystems of an information system. A user terminal **15** is used to input user information into the searching subsystem **22**. A community module **31** allows users to select at least one community **32** which defines a group of users with common interests and a common vocabulary. Examples of groups of people who make up a community might be advertising executives, accountants, higher education counselors, members of a corporation or members of a department within a corporation. Each of these groups of people typically share a specialized professional or organizational vocabulary. The term "lexicon" as used in this application refers to a location in the system memory where search elements expressing the special vocabulary for each group is stored, in order to be used in sophisticated, very targeted searching. Each lexicon typically stores a bank of complex search query specifications **34** using the special vocabulary, or semantic context, of the group. Furthermore, within the lexicon, there may be a library **36** of topics, subtopics, context associations and document types that are of interest to the community, each linked to one or more of the stored search queries.

A profile module **38** may be configured to allow users to choose topics or subtopics from the library of topics **36** which are relevant user's current search needs. Similarly, the user can specify or link additional contextual criteria such as specific geographic locations, industries, or company names. The term "topic" could encompass topics, subtopics or context associations that are listed in the library of topics **36**. The term "topic" may also refer to a document type, like a receivables report or a status report, or to a storage location, such as a web site address. In addition, a user may create her own topic, when she combines a topic with a context, for example. For each user, the profile module is typically configured to write this information to a target profile **40** which is accessed by a query builder module **46** discussed

below, such that the target profile **40** stores the user's list of topics for which automatic searching is desired.

In conjunction with user's target profile and topics specified therein, the present system is typically configured so that the user can specify sources of data to be searched by accessing an atlas module **42** that creates the user atlas **44**. The user atlas specifies the content sources, systems, and locations to be searched by the information system on a regular basis. The user atlas **44** may list internal database locations, external database locations or both.

A query builder module **46** is typically employed to read the target profile. For each entry in the target profile, the query builder module refers to the community lexicon to build a master search query **48** which specifies the information the system must retrieve. By way of example, the master search query **48** may be several search queries associated with several topics linked together or the master search query **48** may be just one search query from the lexicon. A typical process of how the query builder module creates the master search query is described in more detail below. A master search module **50** may be employed to read the master search query and to create autonomous search agents **52** which perform the searches using the criteria in the master search query and at the locations specified in the user atlas. The autonomous search agents use an appropriate format filter for the document locations. Each autonomous search agent searches the data **24** and returns electronic objects that satisfies the search criteria in the master search query. The data **24** may be located internally to the computer system or may be found on external databases that are accessed via communication lines. The electronic objects typically are returned to the master search agent for subsequent indexing and delivery to users.

An indexing subsystem **54** may be employed to create an index of all the terms in an electronic object. Using the index, queries can be run quickly for indexed documents.

There are several different ways an indexing subsystem module **54** could be used. An indexing module could create an index of all terms in all documents residing locally on the system, enabling these locally resident documents to be searched very quickly. In the alternative, indexing can be configured to be performed only when the local documents are identified by the searching subsystem as being relevant. The indexing module may also preferably create a usable index of all terms in each electronic object **56** returned from external sources by the master search module.

Documents that have been indexed are available for easy future use within the community. When an index of a document exists, autonomous search agents **52** typically are not utilized. Rather, the master search query **48** normally operates directly on the index.

A typical indexing module accesses the format filter subsystem **26** and selects the appropriate format filter module **27** to read and write the electronic object to an index table **58**. The index table may contain an abstract of each electronic object and an electronic pointer to where the object is stored. The index terms of all locally stored documents and external documents that have been found in previous searches normally are stored on the index table **58**. In such a configuration, future searching of the indexed documents will be carried out much more quickly than searches of the entire document.

In one preferred embodiment of the system, autonomous search agents are used to search external electronic databases, while the indexing module is used to search local electronic objects and previously retrieved objects from external sources. However, the autonomous agents may be used for all local queries.

When the user chooses an electronic object to view, a retrieval module **60** within a retrieval subsystem **59** is typically employed to access the index table and to select an appropriate format filter module **27** in order to return to display the electronic object to the user in the appropriate display format.

A pattern analysis subsystem **61** may be configured to contain a pattern analysis module **62** that statistically analyzes the information on the index table. Such a pattern analysis module typically produces raw statistics about the placement and frequency of term occurrence. Additional processing may optionally be performed on these statistics, either by the pattern analysis module **62** or by another module designed to use these statistics. For example, in a preferred embodiment of the system, the pattern analysis module **62** will parse the electronic objects on the index table **58** and statistically analyze the appearance of the community lexicon terminology within each electronic object. These object statistics **64** can be used to enhance the community or individual lexicon based on the frequency or infrequency of terms appearing in electronic objects satisfying the search criteria. The object statistics may also be compiled across electronic objects for analysis of subject data, for example, to review patterns of activity in the data, such as merger and acquisition data for specific companies or industries over periods of time. The activities of a typical pattern analysis module **62** will be discussed in greater detail in relation to FIG. **12**.

FIG. **3** shows a more detailed diagram of a typical association between topics **36** and the set of search elements **68** within the community lexicon **34** within a typical topic, there exists one or more search logic atoms representing a complex search query specification developed within the context of the terminology and concerns of the community. Where the user creates a new topic by specifying the combination of two topics, such as specifying a topic and a context association, the query builder module may be configured to concatenate the search atoms for each of the specified topics into a compound search molecule. As an alternative, only one topic might be specified, associated with a search atom. In a typical system, the query builder module passes either the single search atom or the compound search molecule onto the master search module **50**, in the form of a master search query **48**. As described above, the master search module then creates an autonomous search agent for each molecule, or the master search query is used to search indexes of indexed documents. The master search module also may pass information about the search location and appropriate format filter to the autonomous search agent.

FIG. **4** shows how one preferred embodiment forms associations between a topic, subtopic, or context associations, and the corresponding search elements. The example lexicon **32** shown in FIG. **4** is for the Enterprise community, which is a community that may be suited for middle-market company CEOs and their direct reports. Once the user affiliates with the Enterprise community, she can select topics, subtopics, context associations, or report types, in any combination. A particularly useful feature of one embodiment of the present system is the ability of the system to concatenate search query elements into sophisticated, highly specific queries, in order to limit a topic to a specific context, for example. A certain topic, such as a subject or report type, might be limited to a context, such as time, place, specific companies or industries. For

each topic, subtopic, context association or report type, the system maintains a search specification that describes the topic in the syntax of the search agents and index module. The user can also identify other topics which may be of value. The search atoms may be concatenated into a longer, more complex search query. Thus, the search query is a highly refined search tool for the selected topics thereby allowing the system to effectively search and retrieve information that is specific to the needs of the user.

In one mode of operation of the preferred embodiment, the information system can automatically search and retrieve electronic objects relevant to a user or community, and provide a stream of useful information to the user. FIG. **6** outlines how this process is performed. The left half of FIG. **6** shows the general steps that a developer or administrator of the present system may take to prepare the system for use by a specific community. Referring now to FIG. **6**, format filter modules for each document source format are shown to be selected or created in step **80** to allow the system to search electronic objects of various formats. A community atlas is also shown as being created and maintained in step **82** to specify the systems and information resources available to the user or community. Also, the community lexicon **32** may be created and refined at step **84** so that it contains appropriate predefined queries for each topic, subtopic, context association, and report types. Similarly, a default target profile may be created **86** for a hypothetical average user in the community. In this step, the developer may choose topics from the library of topics **36** that are of broad interest to users to be stored in the default target profile. The default target profile typically will be provided to users when they first access the information appliance of the present invention, and can be used until each user creates a personalized target profile. Individual default target profiles may be provided as well.

Once these preliminary steps are performed, the query builder module can combine **88** the search atoms for each default community and individual target profile into complex search queries. Now the system can perform searching steps and stream objects to each user automatically. For each user or community target profile, the master search module may create an autonomous search agent at step **90** to search in the locations specified in the atlas. The autonomous search agent may be configured to select the appropriate format filter module to search documents of varying formats and return the electronic objects at step **92** which satisfy the criteria of the search molecule. In such a configuration, the system then typically checks to determine whether the electronic object has already been indexed for another user or community at step **94**. If the electronic object has already been indexed, the index module may then create an abstract of the electronic object and store the electronic references for the object on the index table at step **98**. If an abstract has already been created and rated, the system may proceed to step **102**. If the electronic object has not already been indexed by the system, the appropriate format filter module may be accessed in step **96** by the index module to create an index of the electronic object retrieved by the autonomous search agent. At step **98**, an abstract typically is created, evaluated and rated **100** for its effectiveness in fulfilling the target profile specifications for each search element for each user. The electronic objects that fulfill the target profile can be organized in a variety of views, for example, according to topic, subtopic, or context association, or report type and streamed to the user on demand at step **102**.

FIG. **7** shows how a user typically retrieves the electronic objects found by the system and personalize the target profile. When accessing the information system for the first time **10**, the user may be presented with a list of high-level topic areas at step **112** from the library of topics **36** that is designed for the community. Pointers and metadata **117** can be configured to store crucial information about content sources such as source location, source type, and other specifications. The user can select topic areas of interest for which electronic objects may be retrieved at step **114**. The user can then select at step **116** the electronic objects of interest. The system typically delivers the first few sentences of each electronic object to the user's viewing frame. The user then selects an electronic object to read at step **118**. That target object may then be streamed in its entirety to the user in step **120** so that the user can view the complete document. If the user would like to view another document at step **122**, she may simply select another object from the list of electronic objects. When the user is finished viewing the found objects, another topic can be selected from the list of topic areas **124**. At step **126**, the user can personalize the target profile or exit the system at step **128**. As discussed in greater detail in relation to FIG. **12**, the system may be configured to keep track of which objects the user and all users choose to view in order to obtain statistical information about the popularity of objects. The continued steps of the user while personalizing the target profile and retrieving electronic objects are shown in FIG. **8**.

FIG. **8** shows a process by which a user can personalize, modify, or refine the target profile. The library of topics may be displayed **130** for each community lexicon associated with the user. The library of topics is stored in the community lexicon **32**. If the user has already created personalized topics, as discussed further below, the user-defined topics may also be displayed at this time. After selecting the topic area of interest **132**, the user typically is presented with a detailed list of subtopics **134** and context associations **136**, where applicable and when available according to the structure of the subject community. Lists of other types may be shown to the user at this point also. Report types or other types of topics may also be presented. Alternatively, the subtopics, context associations, report types, and all other topics may be displayed to the user simultaneously. From the lists, the user can select **138** the subtopics or context associations of interest and select additional links at **140**. The user can further personalize the search string, if desired at step **142**, by creating a free text search string **144**. A personalized search string may be stored in a user lexicon **145** along with other user-designated values. At this point, the user can include **146** another topic area.

The user can also modify **148** the user atlas **44**. The user atlas stores locations of databases where the information appliance will search. This feature allows the user to specify information sources where the search will be most productive and results in a more efficient search by reducing the scope. If the user chooses to change the user atlas **44**, the current atlas settings **150** typically are streamed to the user. Alternatively, for example, the current atlas options may be stored **152** in the target profile. From the list of atlas options, the user can select **154** the databases the system will search. After modifying the atlas, the changes made are stored in the user target profile **152**.

After the system performs a search pursuant to the criteria stored in the user target profile **152**, the user can view **156** the electronic objects retumed by the system. After viewing the document, the user can remain in the system, returning to the list of topics at step **112** in FIG. **7**, to continue to search, or exit the system **128**. In the alternative, after setting up a search target profile, the electronic objects found in the

search could he streamed to the user's E-mail address on a periodic schedule. Returned target documents could also be returned to the user's local hard drive or another storage place on the user's network. This delivery route may be used to allow for perusal while disconnected from remote sources, or to allow the pattern analysis module to operate on the stored retrieved documents.

FIG. **9** shows a more detailed flowchart of the query builder process by which the preferred system may carry out the user-defined search. At scheduled intervals, the query builder module concatenates search atoms associated with the topics in the target profile into search molecules. The process begins at step **180**. The system will then select **182** the first or next user or community target profile **40**. The query builder module will read **184** the next topic from the target profile and identify the appropriate lexicon. If a user has already personalized a target profile, then the system will he accessing the user target profile at this time. However, if the user has not yet created a personalized target profile, the default community target profile will be accessed.

In the emhodiment shown in FIG. **9**, the search query for that particular topic is read **186** and placed in temporary storage. The query huilder module will then determine if there exists a context association **188** or other relevant topics or subtopics to be combined for the specified topic or subtopic. If no context association or other topic or subtopic exists, the search molecule **194** is complete. Otherwise, the query huilder module will read and concatenate **190** the search atoms for the context association or other relevant topics or suhtopics to those already in temporary storage. This concatenation process will continue **192** until no further context associations or other relevant topics or subtopics exists. Once the concatenation process is completed, the concatenated search molecule is availahle for use by the master search module **50**, shown in FIG. **5**. If there is another unprocessed topic **196** in the current target profile, the process starting at step **184** is repeated. Similarly, the query huilder module checks for unprocessed users or community target profiles **198**. The system returns to step **182** to process additional users or communities. If these do not exist, the query huilder process **199** will end until the next scheduled iteration or user-initiated search.

FIG. **10** shows additional detail of the indexing process. The indexing module is shown to be started **200** either when a query is made against the indexing module or when an electronic ohject is returned by an autonomous search agent to the master search module for storage. As discussed above with reference to FIG. **5**, there are many different possible ways for the indexing module to operate on internal documents, external documents or both. Assuming that all internal electronic objects will be indexed, the indexing module reads the storage references **202** written by the master search agent to see which servers, directories, and databases have material for indexing. The indexing module then determines **204** whether the format or structure of the electronic object is that of a file or datahase.

In the process shown in FIG. **10**, if the structure is a file, the indexing module reads the file extension, header, and initial bytes of the file **206** to determine the file format. Thus, the appropriate format filter module can be selected. The indexing module then determines whether the ohject has been written to the indexing table **208** since the previous indexing module run If it has not heen updated since an index was last created or the object has never been indexed, the next file is read and its format is determined for indexing. Otherwise, the indexing module will access **210** the appropriate filter module so that it can read the document and create a new full-text index and ahstract for the updated electronic object.

If the format or structure of the electronic object is that of a database, the indexing module typically negotiates **212** the database's security and access the database. Next, the indexing module will select **214** the appropriate format filter module to format the electronic object for indexing. Finally, the indexing module will check to ensure that the electronic object has not already been written to the indexing table **216**. After the indexing module has read the object and created an index and abstract of the electronic object **210**, the indexing module stores **218** a full-text index ahstract and location reference for the electronic object.

The indexing module typically ascertains whether the electronic object is a file or datahase server at step **220**. If more files in a directory structure **222** or more objects in the datahase **224** still remain, the process is repeated starting at step **206** or step **214** respectively. Furthermore, if another datahase exists **226**, the process of negotiating the security and accessing that database is continued starting at step **212**. Finally, the indexing module will determine whether or not another server needs to be indexed **228**. If so, the process of reading the storage references in the system will he repeated starting at step **202**. If no more servers need to be indexed, the indexing module is terminated **230** until the next iterative cycle.

FIG. **11** shows how a lexicon may be developed for a specific community. First, a specific audience typically is identified **252** based upon the business rules or other frame of reference common to the audience. Memhers of the target audience can be interviewed **254** to determine the a) types and sources of new topics of interest to the community; h) the types and sources of learning and business performance improvement subjects of interest to the community; c) the types and sources of technical suhjects of interest to the community; d) types and sources of financial and business management systems reports of interest to the targeted community; and e) the types and sources of other business documents and on-line discussion topics or subjects of interest to the community. Based on these interviews, the topic areas of highest priority **256** typically are identified. These interviews will also identify important data locations that may be made available for selection in the atlas. Lexicon development may proceed by reviewing **258** the vocahulary that applies to this audience, for example hy referring to professional dictionaries or articles and by drafting **260** a high-level framework of the topic areas for the community. Where required, this high-level framework is further broken down into subcategories **262** for the topic area. The lexicon developed to this point should be tested **264** using focus groups to ensure that the terminology is within the framework of the topic concepts used hy the community.

In developing the lexicon, another important aspect of the preferred embodiment is to ensure that topic areas are separated **266** into stand-alone lists where possible, such as industries, geographic locations, and company names. This serves to minimize the hierarchical relationships and maximize the many-to-many relationships for the query builder to concatenate. Predefined search queries **268** should be created for each topic, subtopic, context association, or report type utilizing the linguistic context of the community and the desired information resources that will he part of the system. These predefined search atoms should be tested **270** against appropriate content and refined accordingly. Similar to the predefined search topics and subtopics, each element of each free-standing context list typically is defined in the

search elements within the community lexicon, and wherein each topic identifies a subject or concept of interest that is relevant to the information needs of the community.

22. The method of claim **21**, the method further comprising creating a target profile by selecting at least one topic from the library of topics.

23. The method of claim **19**, the method further comprising creating a user atlas by selecting at least one preferred data resource from a list of data resources from which objects may be retrieved.

24. The method of claim **19**, the method further comprising:

creating a target profile by selecting at least one topic from a library of topics, wherein each topic is associated with one or more of the predefined search elements and each topic identifies a subject that is relevant to the information needs of a community of users; and

creating an electronic master search query by concatenating the search elements associated with each topic listed in the target profile.

25. The method of claim **24**, wherein the electronic master search query is used to search the object.

26. The method of claim **19**, wherein the step of searching the object is scheduled to occur automatically at specified time intervals.

27. The method of claim **19**, further comprising creating an index of an object that was identified in the searching step, by compiling and storing in computer readable medium summary information that identifies the object.

28. The method of claim **19**, further comprising sifting through the objects identified in the searching step to recognize and count words within each object that are in the lexicon.

29. The method of claim **19**, further comprising locating terms within the identified objects according to frequency and location of the terms in relation to words in each object that are in the lexicon.

30. The method of claim **19**, further comprising:

retrieving objects that are identified in the searching step;

recording a number of times that each object has been identified; and

reporting the number of times that each object has been identified.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 6,029,165
DATED : February 22, 2000
INVENTOR(S) : Gable

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Column 2,
Line 15, insert the heading -- SUMMARY OF THE INVENTION --

Column 6,
Line 21, "Microsoft" should read -- Microsoft® --

Column 10,
Line 2, "10" should read -- 110 --

Signed and Sealed this

Eighteenth Day of September, 2001

Attest:

Nicholas P. Godici

Attesting Officer

NICHOLAS P. GODICI
Acting Director of the United States Patent and Trademark Office

US006526400B1

(12) **United States Patent**
Takata et al.

(10) **Patent No.: US 6,526,400 B1**
(45) **Date of Patent: Feb. 25, 2003**

(54) **INFORMATION SEARCH APPARATUS AND METHOD**

(75) Inventors: **Tomomi Takata**, Yokohama (JP); **Yuji Kobayashi**, Kawasaki (JP)

(73) Assignee: **Canon Kabushiki Kaisha**, Tokyo (JP)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/409,090**

(22) Filed: **Sep. 30, 1999**

(30) **Foreign Application Priority Data**

Sep. 30, 1998 (JP) 10-278723
Sep. 30, 1998 (JP) 10-278725

(51) **Int. Cl.**[7] **G06F 17/30**
(52) **U.S. Cl.** **707/3**; 707/2; 707/4; 707/5; 707/6; 707/10
(58) **Field of Search** 707/3, 2, 4, 5, 707/10, 6

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,778,368	A		7/1998	Hogan et al. 707/10
5,842,206	A		11/1998	Sotomayor 707/5
5,884,309	A		3/1999	Vanechanos, Jr. 707/10
5,893,092	A		4/1999	Driscoll 707/5
5,893,095	A	*	4/1999	Jain et al. 707/6
5,983,237	A	*	11/1999	Jain et al. 707/104
6,078,916	A		6/2000	Culliss 707/5
6,085,186	A		7/2000	Christianson et al. 707/3
6,088,692	A		7/2000	Driscoll 707/5
6,102,969	A		8/2000	Christianson et al. 717/8
6,122,628	A		9/2000	Castelli et al. 707/5
6,122,647	A		9/2000	Horowitz et al. 707/513
6,175,829	B1		1/2001	Li et al. 707/3
6,182,068	B1		1/2001	Culliss 707/5
6,226,636	B1	*	5/2001	Adel-Mottaleb et al. 707/4
6,249,784	B1		6/2001	Macke et al. 707/3
6,347,313	B1		2/2002	Ma et al. 707/3

OTHER PUBLICATIONS

Mukherjea, Sougata et al., "Visualizing World-Wide Web Search Engine Results", Proceedings of the 1999 IEEE International Conference on Information Visualization, Jul. 14–16, 1999, pp. 400–405.

Yanai, Keiji, "An Automatic Image-Gathering System for the World-Wide Web by Integration of Keywords and Image Features", Proceedings, Fourth International Conference on Computational Intelligence and Multimedia Applications, Oct. 2001, pp. 303–307.

Chung-Sheng Li, Mohan, R., Smith, J.R., Multimedia Content description in the InfoPyramid,. Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing Volume; 6, May 12–15, 1998, pp.: 3789–3792.

Hjelsvold, R., Midtsraum, R., Sandsta, O., Searching and browsing a Shared Video Database, Proceedings., International Workshop on Multi-Media Database Management Systems, 1995., Aug. 28–30, 1995, pp.: 90–98.

Wittenburg, T.M., Little, T.D.C., An adaptive document management System for Shared Multimedia Data. Proceedings of the International Conference on Multimedia Computing and Systems, 1994., May 15–19, 1994; pp.: 245–254.

* cited by examiner

Primary Examiner—Frantz Coby
(74) *Attorney, Agent, or Firm*—Fitzpatrick, Cella, Harper & Scinto

(57) **ABSTRACT**

An image search apparatus acquires associative words in relation to an input query word, and makes a keyword search of image information on the basis of the obtained associative words and input query word. Parallel to the keyword search, the apparatus acquires a query word and sensory patterns corresponding to the query word, and makes a feature amount search using the feature amounts of the obtained sensory patterns. A search result integration process integrates these search results at weight ratios which are set in advance to obtain final search results. In this manner, the multimedia information wanted can be accurately extracted with respect to the input query word.

23 Claims, 22 Drawing Sheets

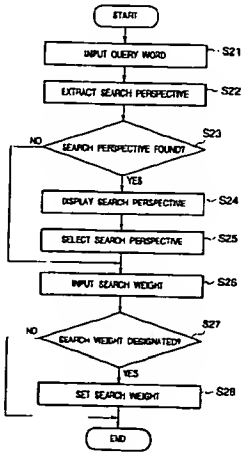


FIG. 1

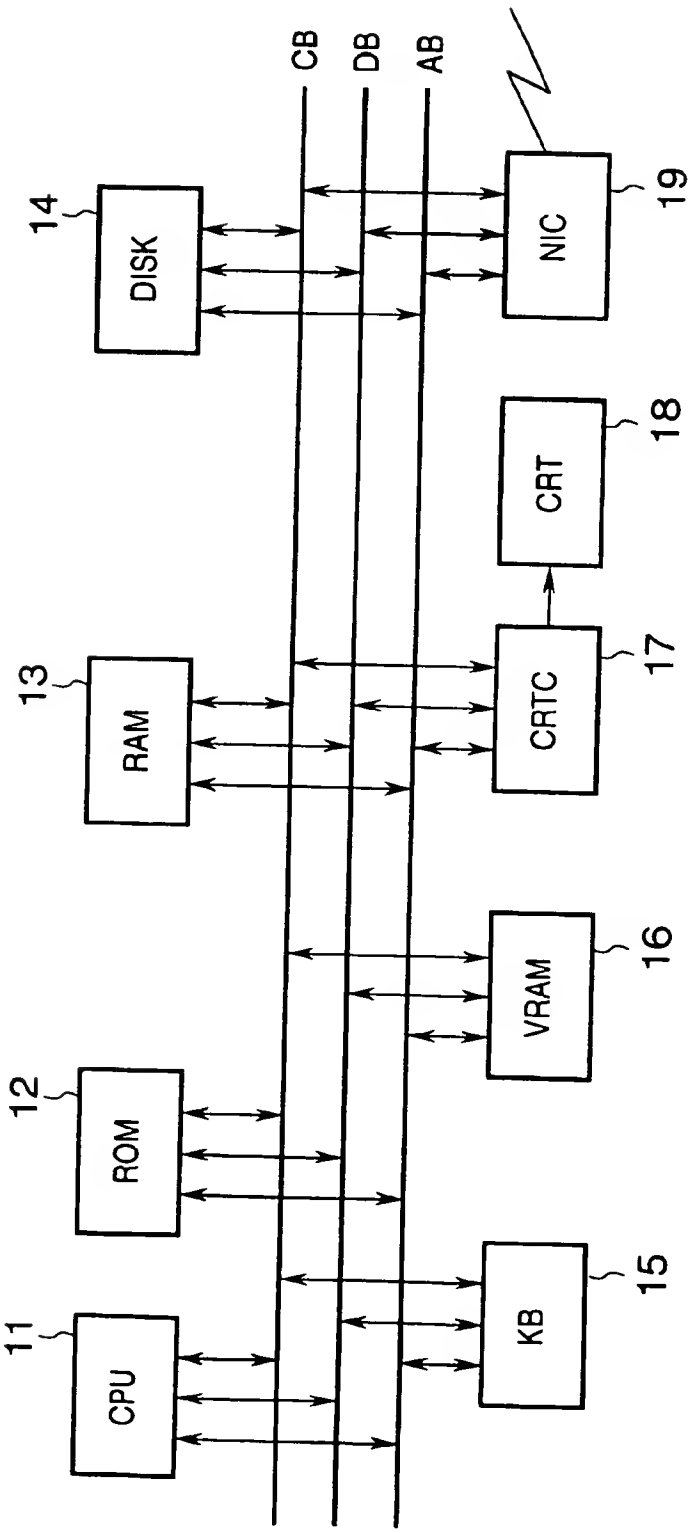

CB
DB
AB
14
DISK
19
NIC
18
CRT
13
RAM
17
CRTC
16
VRAM
12
ROM
15
KB
11
CPU

FIG. 2

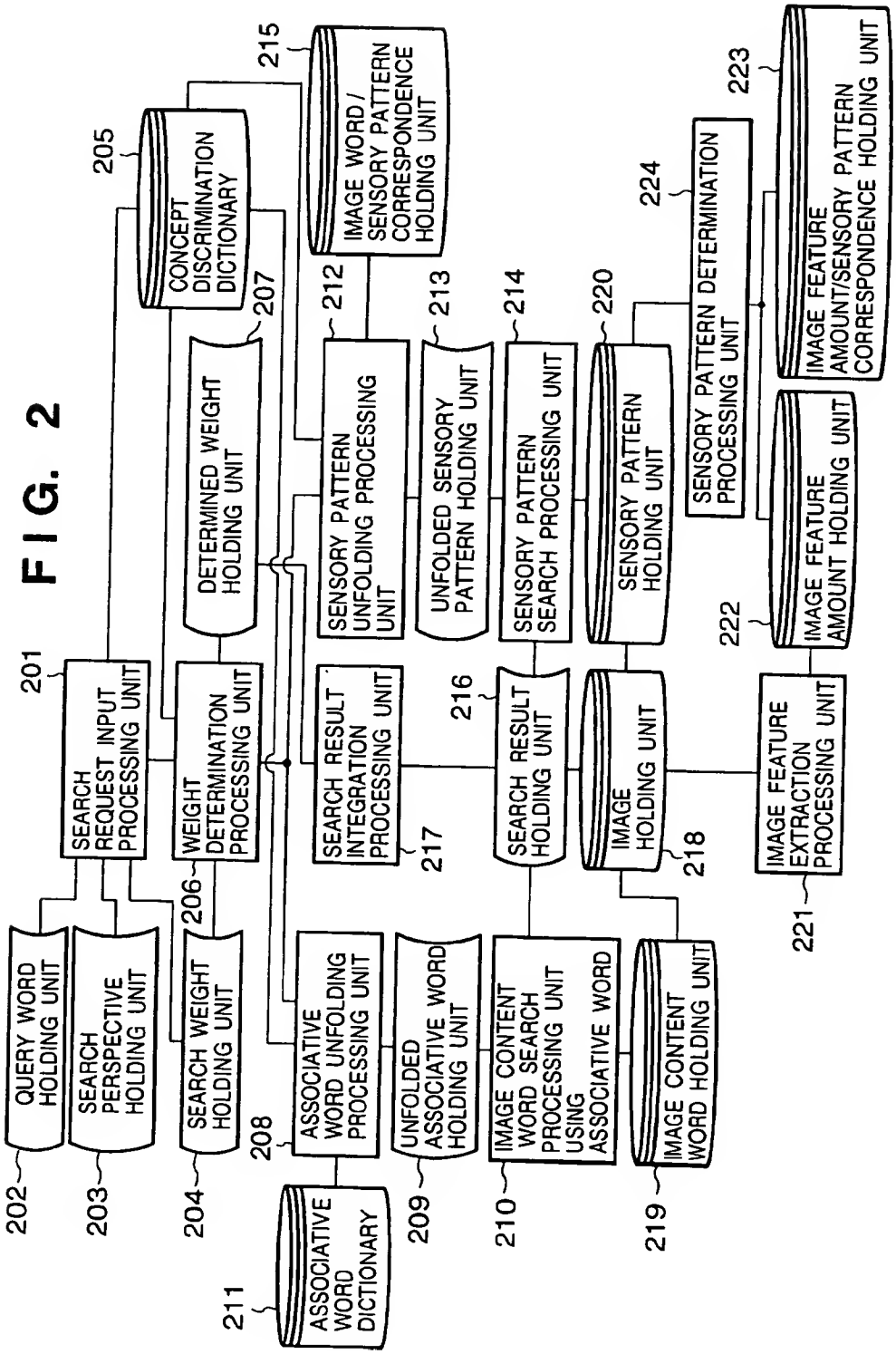

SEARCH REQUEST INPUT PROCESSING UNIT
201
QUERY WORD HOLDING UNIT
202
SEARCH PERSPECTIVE HOLDING UNIT
203
SEARCH WEIGHT HOLDING UNIT
204
206
CONCEPT DISCRIMINATION DICTIONARY
205
DETERMINED WEIGHT HOLDING UNIT
207
WEIGHT DETERMINATION PROCESSING UNIT
ASSOCIATIVE WORD UNFOLDING PROCESSING UNIT
208
UNFOLDED ASSOCIATIVE WORD HOLDING UNIT
209
ASSOCIATIVE WORD DICTIONARY
211
IMAGE WORD / SENSORY PATTERN CORRESPONDENCE HOLDING UNIT
215
SENSORY PATTERN UNFOLDING PROCESSING UNIT
212
UNFOLDED SENSORY PATTERN HOLDING UNIT
213
SEARCH RESULT INTEGRATION PROCESSING UNIT
217
SENSORY PATTERN SEARCH PROCESSING UNIT
214
SEARCH RESULT HOLDING UNIT
216
IMAGE CONTENT WORD SEARCH PROCESSING UNIT USING ASSOCIATIVE WORD
210
IMAGE CONTENT WORD HOLDING UNIT
219
SENSORY PATTERN HOLDING UNIT
220
IMAGE HOLDING UNIT
218
SENSORY PATTERN DETERMINATION PROCESSING UNIT
224
IMAGE FEATURE EXTRACTION PROCESSING UNIT
221
IMAGE FEATURE AMOUNT HOLDING UNIT
222
IMAGE FEATURE AMOUNT/SENSORY PATTERN CORRESPONDENCE HOLDING UNIT
223

FIG. 3

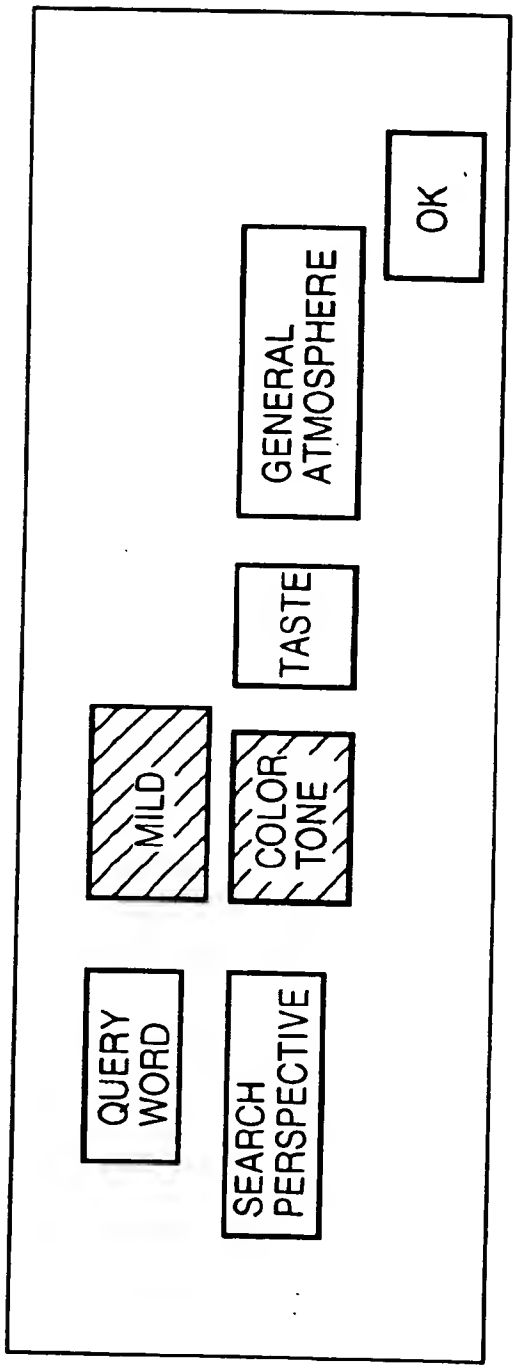

OK
GENERAL ATMOSPHERE
TASTE
COLOR TONE
MILD
SEARCH PERSPECTIVE
QUERY WORD

FIG. 4

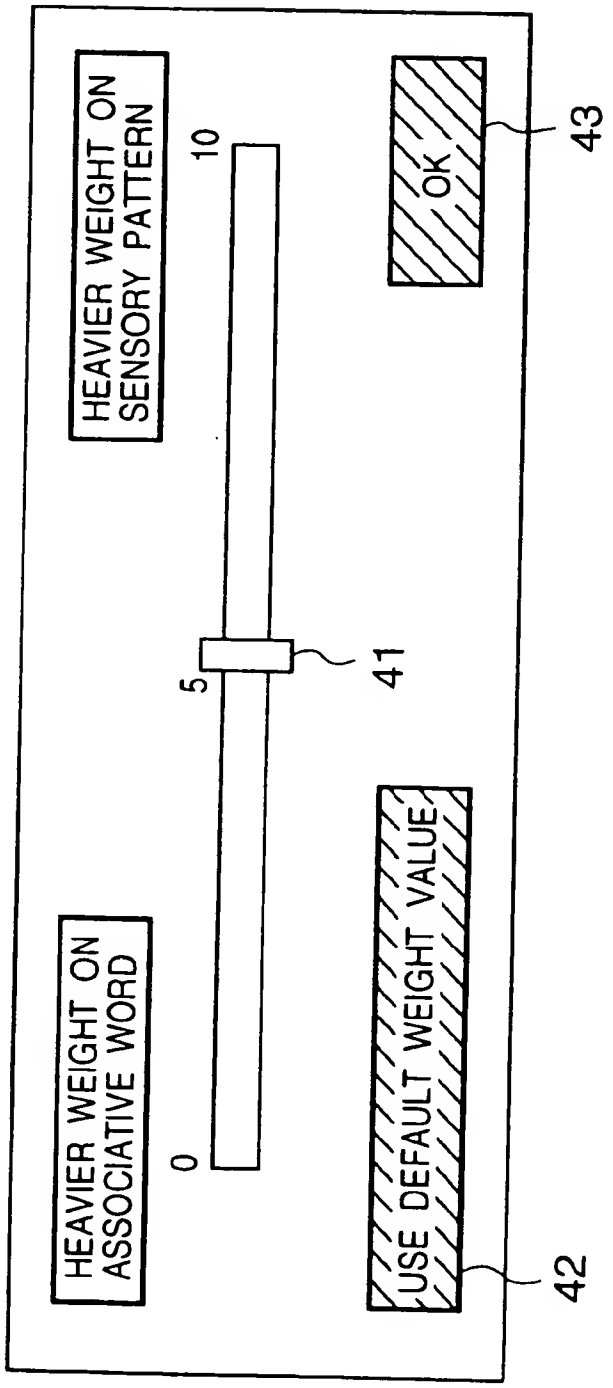

HEAVIER WEIGHT ON SENSORY PATTERN
10
OK
43
5
41
HEAVIER WEIGHT ON ASSOCIATIVE WORD
0
USE DEFAULT WEIGHT VALUE
42

FIG. 5

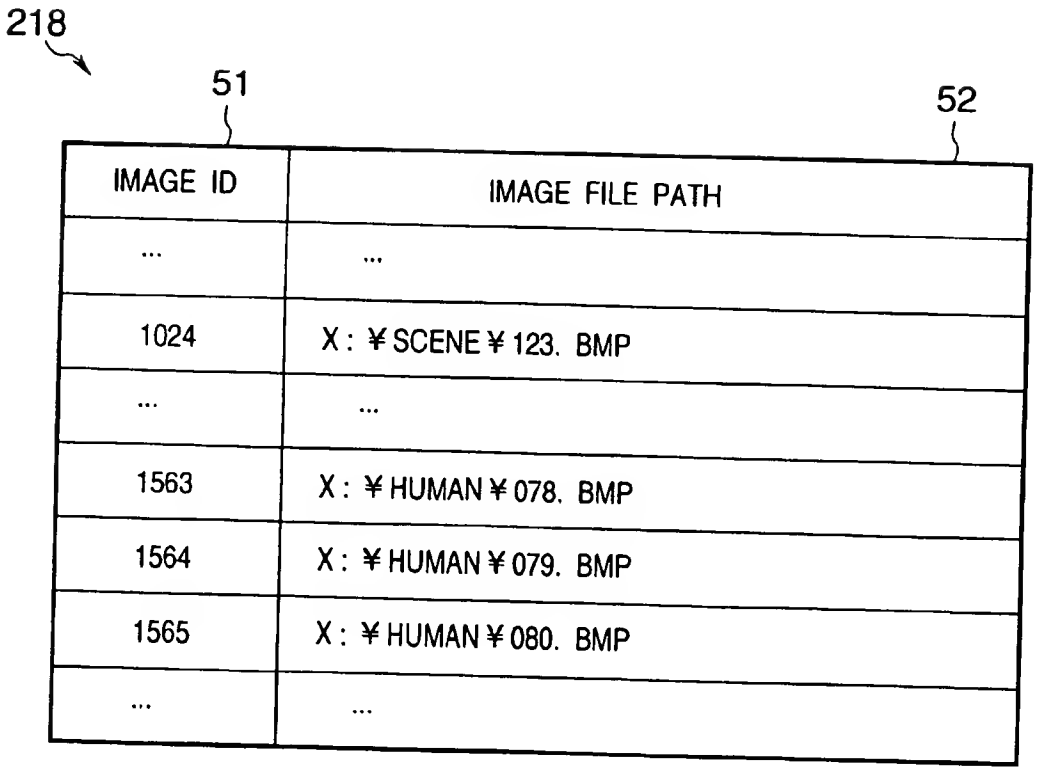


IMAGE ID	IMAGE FILE PATH
...	...
1024	X : ¥SCENE¥123. BMP
...	...
1563	X : ¥HUMAN¥078. BMP
1564	X : ¥HUMAN¥079. BMP
1565	X : ¥HUMAN¥080. BMP
...	...

FIG. 6

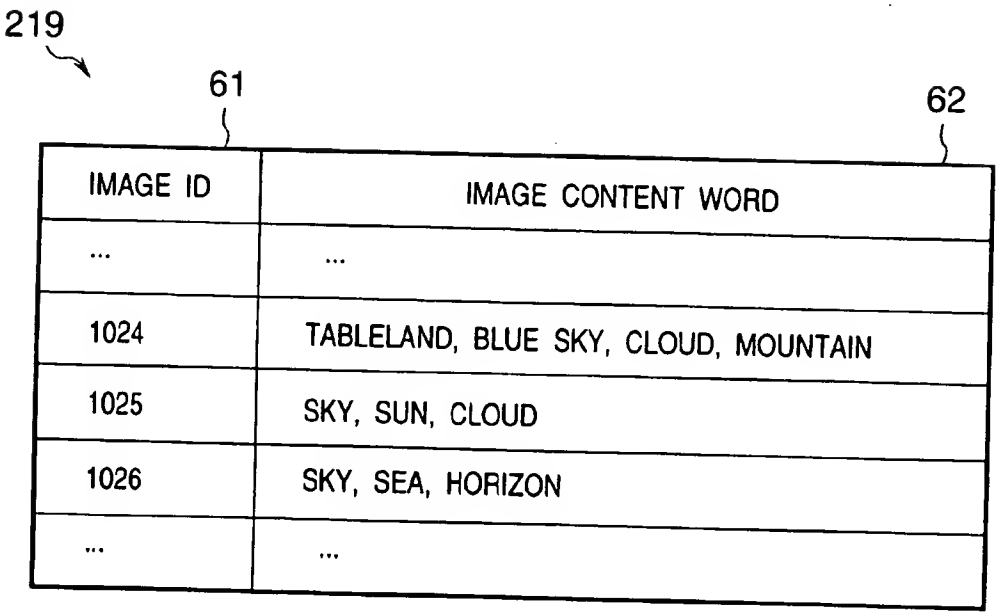

219
61
62
IMAGE ID
IMAGE CONTENT WORD
...
...
1024
TABLELAND, BLUE SKY, CLOUD, MOUNTAIN
1025
SKY, SUN, CLOUD
1026
SKY, SEA, HORIZON
...
...

FIG. 7

71 72

IMAGE CONTENT WORD	IMAGE ID
...	...
BLUE SKY	988, 1020, 1024
SEA	867, 888, 1026, 1347
CLOUD	991, 1024, 1025, 1078
TABLELAND	1024
SKY	777, 778, 1025, 1026
SUN	1025, 1066
MOUNTAIN	1024, 1111, 1112, 1113, 1115
...	...

FIG. 8

205

INDEX WORD (80)	SEARCH PERSPECTIVE (81)	ANTITHETIC CONCEPT (82)	ASSOCIATED WEIGHT (83)	SENSORY PATTERN WEIGHT (84)
...	...	...	...	...
REFRESHING	CHARACTER DESCRIPTION	SHABBY	8	2
REFRESHING	LANDSCAPE	HEAVY	3	7
REFRESHING	TASTE	THICK	9	1
REFRESHING	GENERAL ATMOSPHERE		5	5
...	...	...	...	...
MILD	COLOR TONE	SHOWY	2	8
MILD	TASTE		9	1
MILD	GENERAL ATMOSPHERE		5	5
...	...	...	...	...
SIMPLE	GENERAL ATMOSPHERE	URBAN	5	5
SIMPLE	COLOR TONE	SHOWY	3	7
...	...	...	...	...
SMOOTH	FEEL	SANDY	7	3
SMOOTH	TEXTURE	SANDY	5	5
SMOOTH	GENERAL ATMOSPHERE	SANDY	6	4
...	...	...	...	...

FIG. 9

211

90 ASSOCIATIVE ID	91 INDEX WORD	92 ASSOCIATIVE WORD	93 ASSOCIATIVE PERSPECTIVE	94 ASSOCIATION STRENGTH
...	...	...	...	...
58790	REFRESHING	ATHLETE	CHARACTER DESCRIPTION	6
58791	REFRESHING	NICE GUY	CHARACTER DESCRIPTION	5
58792	REFRESHING	SUMMER RESORT	LANDSCAPE	6
58793	REFRESHING	TABLELAND	LANDSCAPE	7
58794	REFRESHING	BLUE SKY	LANDSCAPE	8
58795	REFRESHING	LEMON	TASTE	6
58796	REFRESHING	SODA	TASTE	4
58797	REFRESHING	SHOWER	GENERAL ATMOSPHERE	5
...	...	...	...	...
71238	MILD	TEA	TASTE	7
71239	MILD	WINE	TASTE	2
71240	MILD	EARTH COLOR	COLOR TONE	4
...	...	...	...	...
126531	SIMPLE	COUNTRY	GENERAL ATMOSPHERE	5
126532	SIMPLE	LOCAL STYLE COOKING	GENERAL ATMOSPHERE	5
126533	SIMPLE	FOLKCRAFT ARTICLE	GENERAL ATMOSPHERE	6
126534	SIMPLE	SPACE SHUTTLE	GENERAL ATMOSPHERE	−8
126535	SIMPLE	SPORT CAR	GENERAL ATMOSPHERE	−8
126536	SIMPLE	CHANDELIER	GENERAL ATMOSPHERE	−9
...	...	...	...	...
234099	SMOOTH	VELVETY SKIN	FEEL	3
234100	SMOOTH	VELVET	FEEL	6
234101	SMOOTH	LUSTER	TEXTURE	5
...	...	...	...	...

FIG. 10

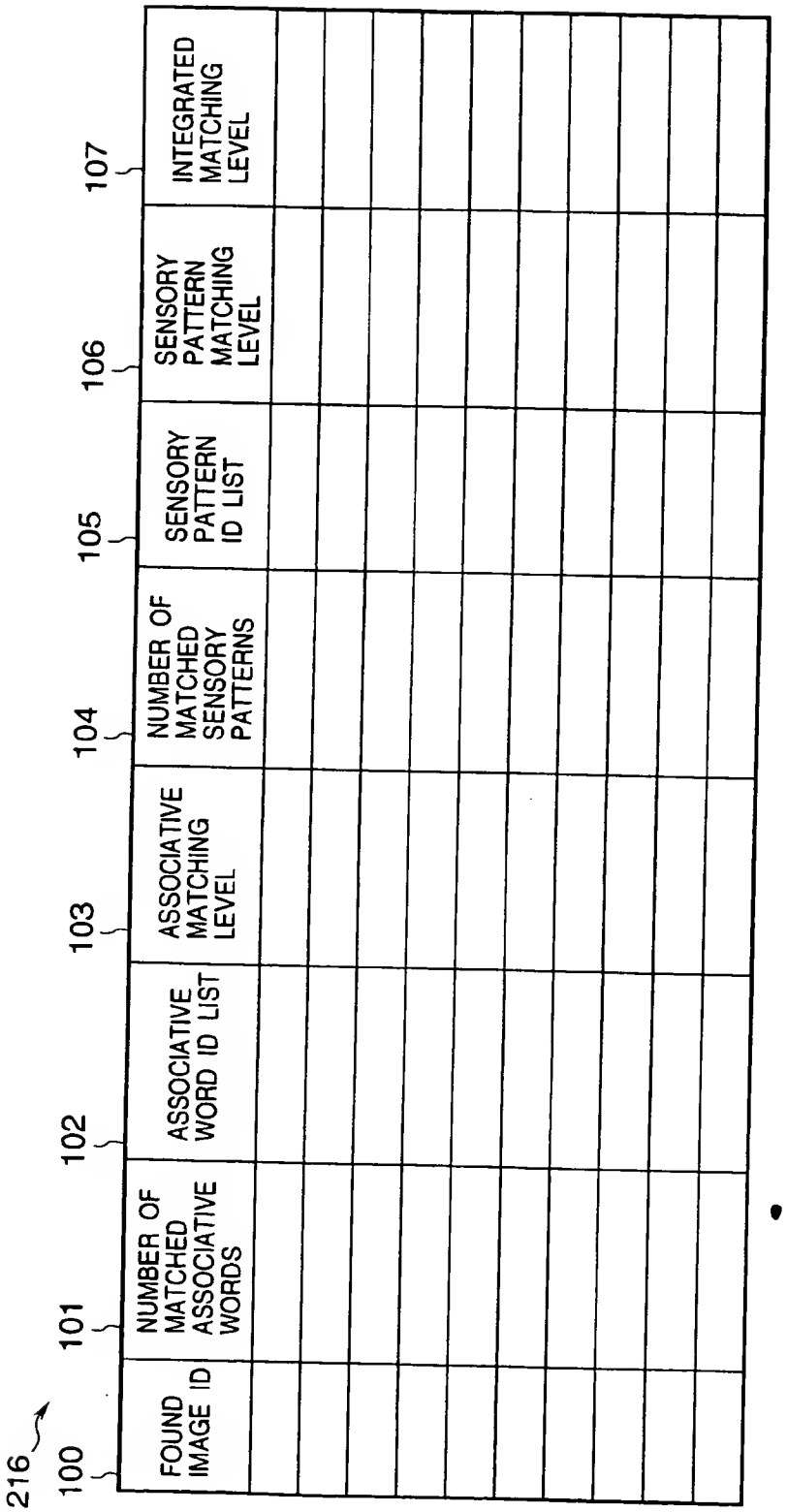

216
100
FOUND IMAGE ID
101
NUMBER OF MATCHED ASSOCIATIVE WORDS
102
ASSOCIATIVE WORD ID LIST
103
ASSOCIATIVE MATCHING LEVEL
104
NUMBER OF MATCHED SENSORY PATTERNS
105
SENSORY PATTERN ID LIST
106
SENSORY PATTERN MATCHING LEVEL
107
INTEGRATED MATCHING LEVEL

FIG. 11

IMAGE WORD (2130-1)	R E F R E S H I N G NULL ...
NUMBER OF SENSORY PATTERNS (2130-2)	7
STORAGE LOCATION ADDRESS OF PATTERN INFORMATION (2130-3)	

2130-4

SENSORY PATTERN ID (2130-5)	5
ASSOCIATIVE WORD (2130-6)	F O R E S T NULL ...
SENSORY PATTERN ID	12
ASSOCIATIVE WORD	B L U E S K Y NULL ...
...	...

FIG. 12

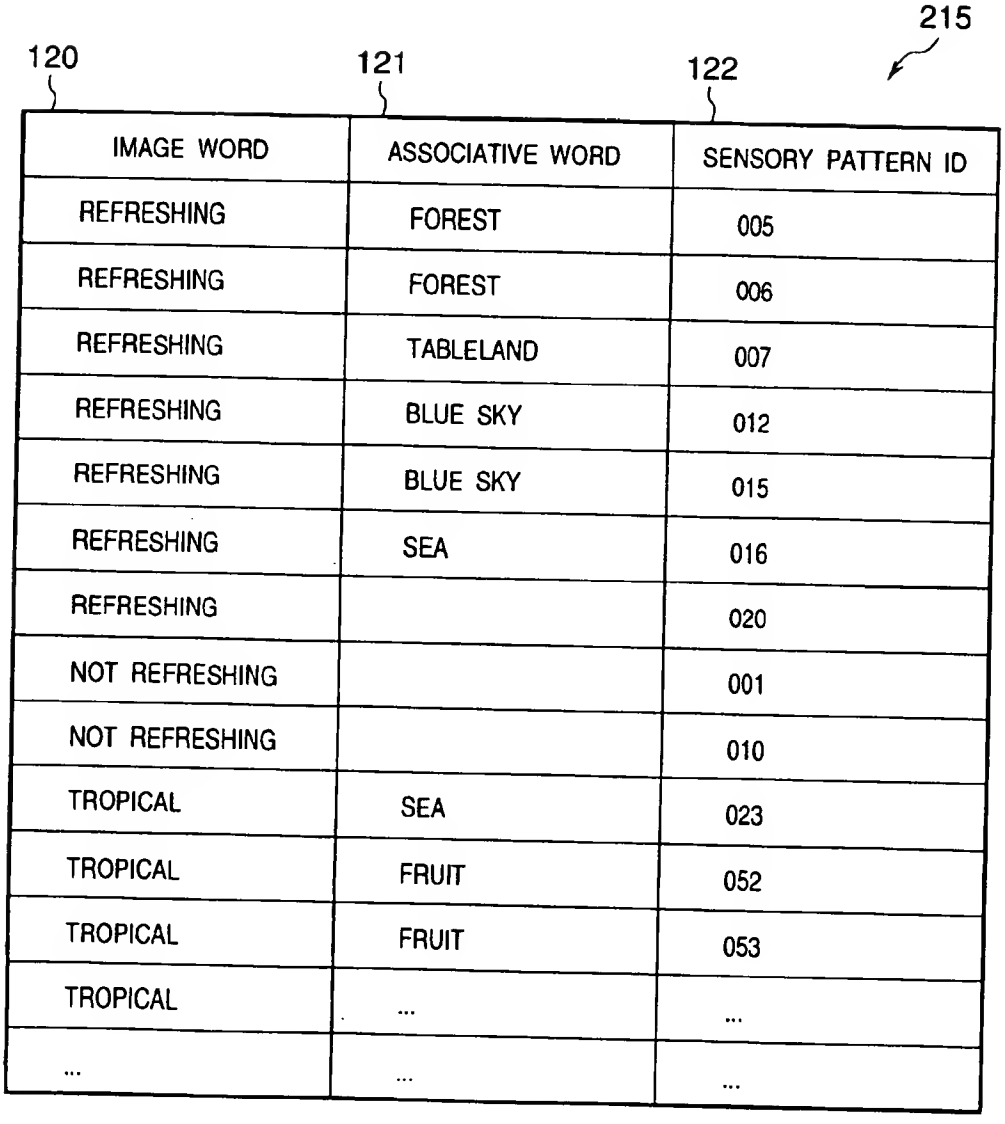


IMAGE WORD	ASSOCIATIVE WORD	SENSORY PATTERN ID
REFRESHING	FOREST	005
REFRESHING	FOREST	006
REFRESHING	TABLELAND	007
REFRESHING	BLUE SKY	012
REFRESHING	BLUE SKY	015
REFRESHING	SEA	016
REFRESHING		020
NOT REFRESHING		001
NOT REFRESHING		010
TROPICAL	SEA	023
TROPICAL	FRUIT	052
TROPICAL	FRUIT	053
TROPICAL	...	...
...	...	...

FIG. 13

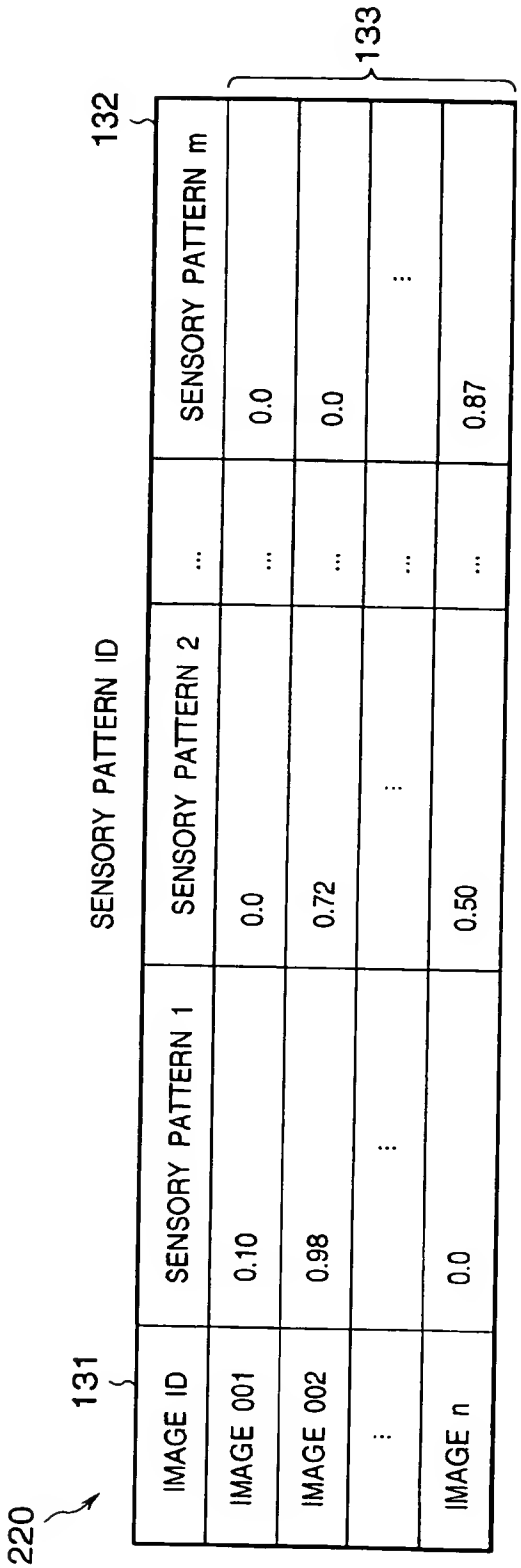

220
131
132
133
SENSORY PATTERN ID
IMAGE ID
SENSORY PATTERN 1
SENSORY PATTERN 2
...
SENSORY PATTERN m
IMAGE 001
0.10
0.0
...
0.0
IMAGE 002
0.98
0.72
...
0.0
...
...
...
...
...
IMAGE n
0.0
0.50
...
0.87

FIG. 14

PHYSICAL IMAGE FEATURE (X_1 ~X_n)

	X_1	X_2	X_3	X_4	X_5	X_6	···	Xn
B1	x11	x12	X13	X14	X15	X16		X1n
B2	x21	x22	x23	x24	x25	x26		x2n
B3	x31	x32	x33	x34	x35	x36		x3n
B4	x41	x42	x43	x44	x45	x46		x4n
B5	x51	x52	x53	x54	x55	x56		x5n
⋮								
Bm	xm1	xm2	xm3	xm4	xm5	xm6		xmn

B1···Bm : REGION OR BLOCK NO.

FIG. 15

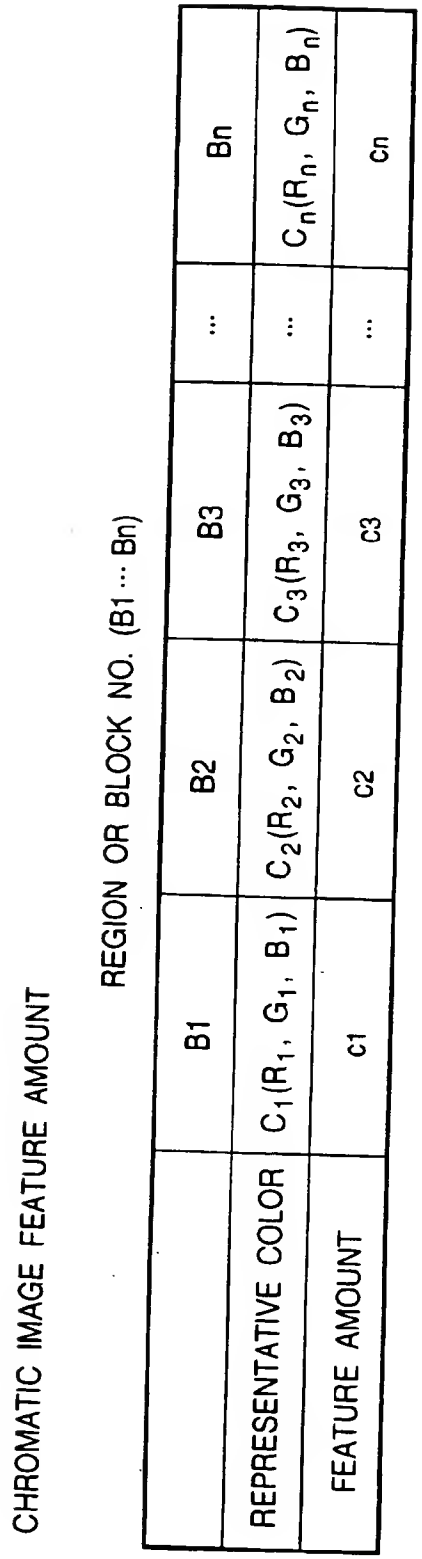

CHROMATIC IMAGE FEATURE AMOUNT
REGION OR BLOCK NO. (B1 ··· Bn)
B1
B2
B3
···
Bn
REPRESENTATIVE COLOR
$C_1(R_1, G_1, B_1)$
$C_2(R_2, G_2, B_2)$
$C_3(R_3, G_3, B_3)$
···
$C_n(R_n, G_n, B_n)$
FEATURE AMOUNT
c1
c2
c3
···
cn

FIG. 16

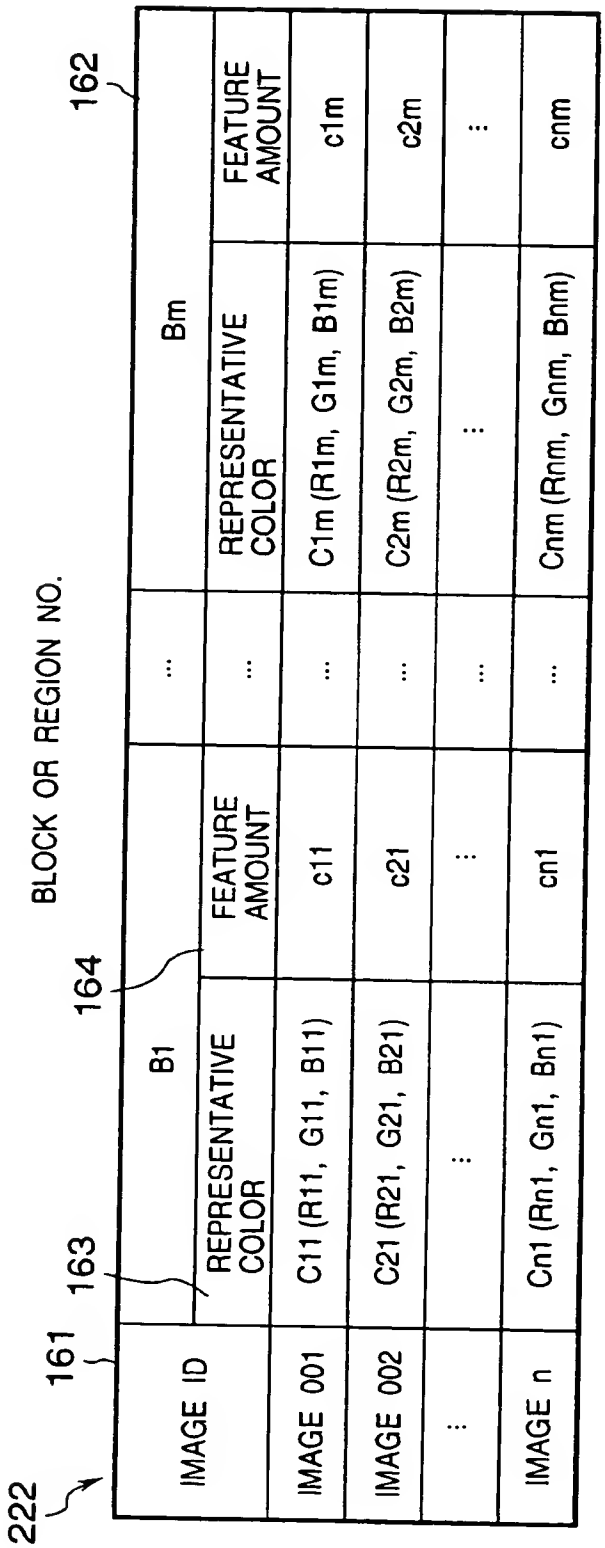

222
161
163
164
162
BLOCK OR REGION NO.
IMAGE ID
B1
Bm
REPRESENTATIVE COLOR
FEATURE AMOUNT
IMAGE 001
C11 (R11, G11, B11)
c11
C1m (R1m, G1m, B1m)
c1m
IMAGE 002
C21 (R21, G21, B21)
c21
C2m (R2m, G2m, B2m)
c2m
IMAGE n
Cn1 (Rn1, Gn1, Bn1)
cn1
Cnm (Rnm, Gnm, Bnm)
cnm

FIG. 17

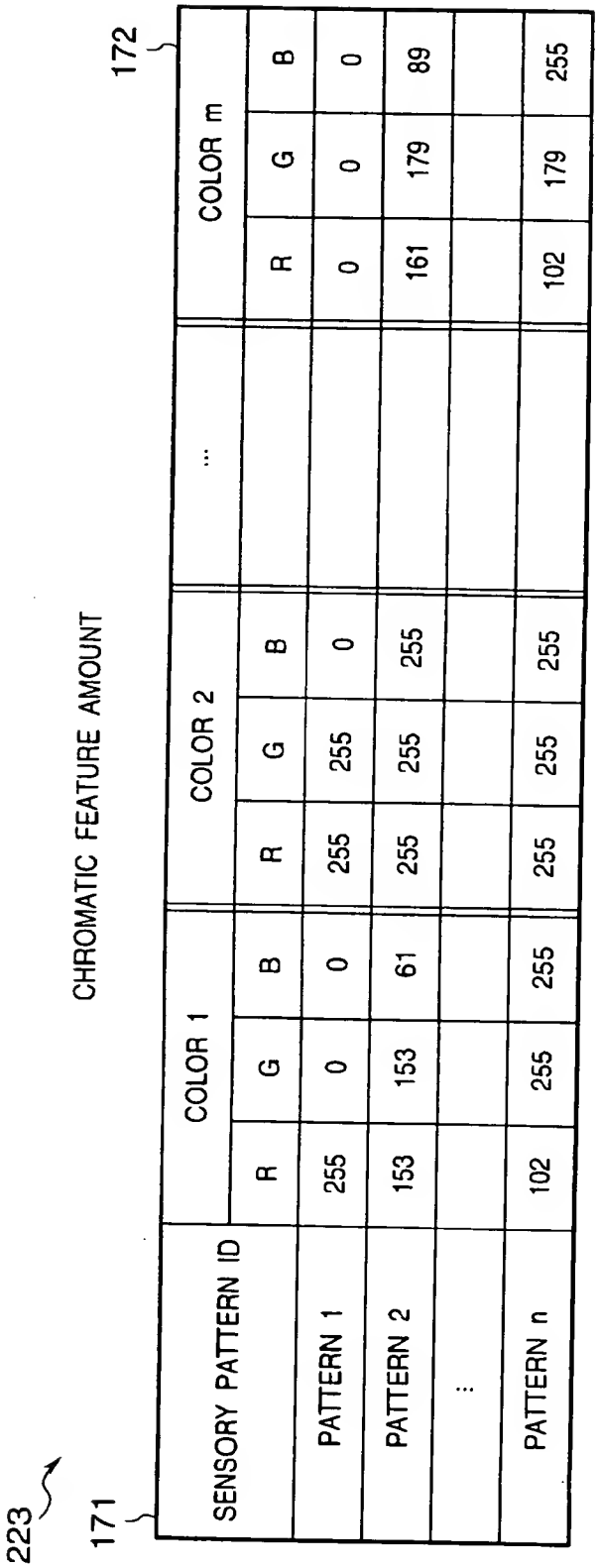

223
171
172
CHROMATIC FEATURE AMOUNT
SENSORY PATTERN ID
COLOR 1
COLOR 2
...
COLOR m
R
G
B
PATTERN 1
255
0
0
255
255
0
0
0
0
PATTERN 2
153
153
61
255
255
255
161
179
89
...
PATTERN n
102
255
255
255
255
255
102
179
255

FIG. 18

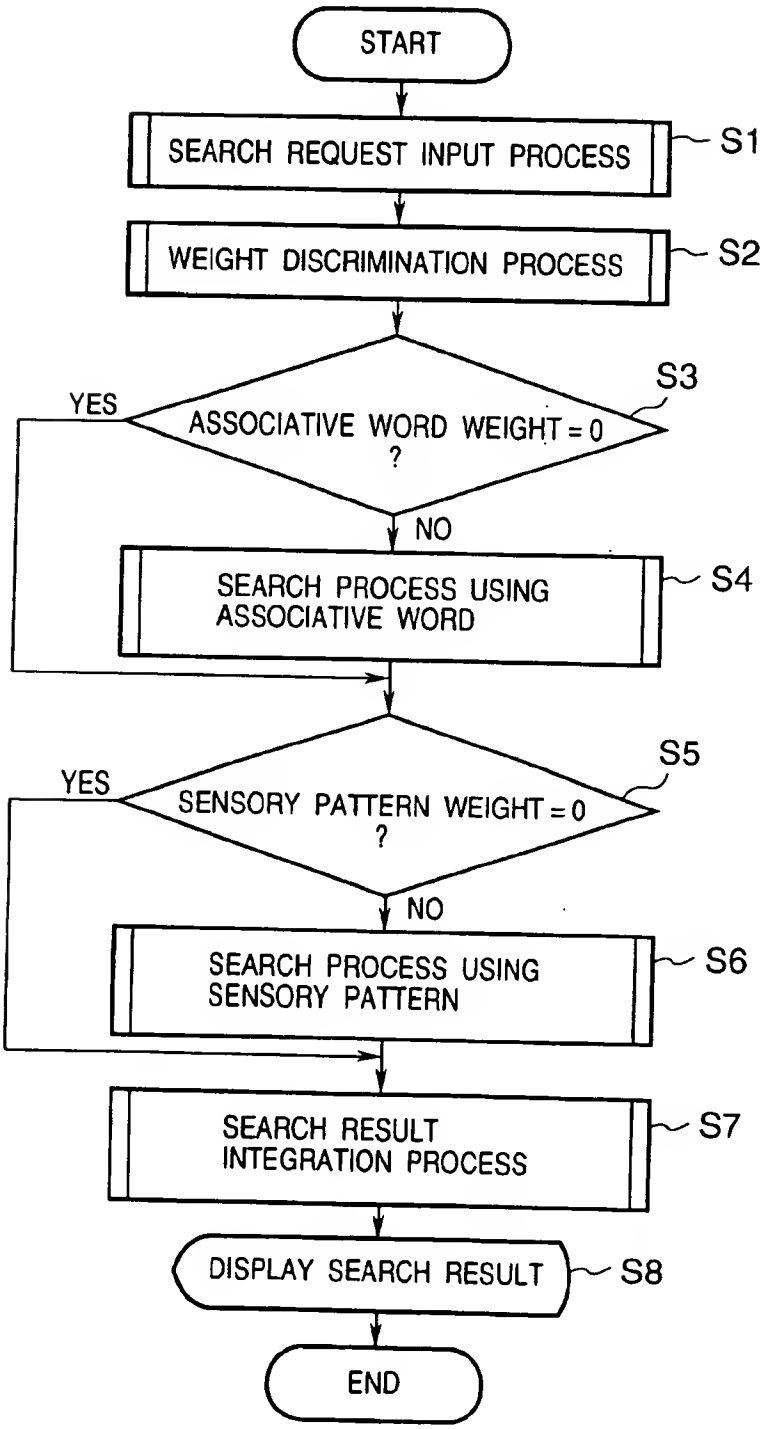

START
SEARCH REQUEST INPUT PROCESS
S1
WEIGHT DISCRIMINATION PROCESS
S2
S3
YES
ASSOCIATIVE WORD WEIGHT = 0 ?
NO
SEARCH PROCESS USING ASSOCIATIVE WORD
S4
S5
YES
SENSORY PATTERN WEIGHT = 0 ?
NO
SEARCH PROCESS USING SENSORY PATTERN
S6
SEARCH RESULT INTEGRATION PROCESS
S7
DISPLAY SEARCH RESULT
S8
END

FIG. 19

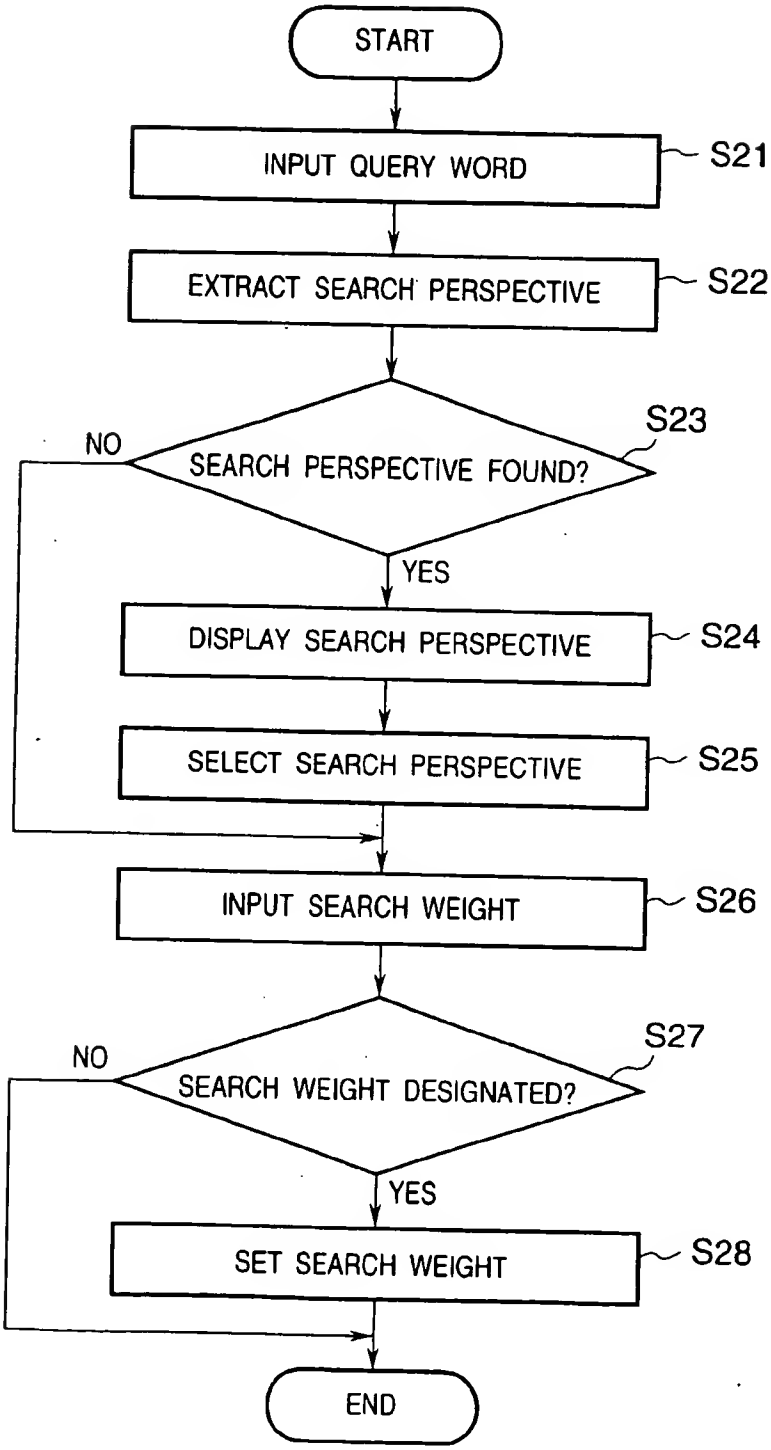

START
INPUT QUERY WORD
S21
EXTRACT SEARCH PERSPECTIVE
S22
SEARCH PERSPECTIVE FOUND?
S23
NO
YES
DISPLAY SEARCH PERSPECTIVE
S24
SELECT SEARCH PERSPECTIVE
S25
INPUT SEARCH WEIGHT
S26
SEARCH WEIGHT DESIGNATED?
S27
NO
YES
SET SEARCH WEIGHT
S28
END

FIG. 20

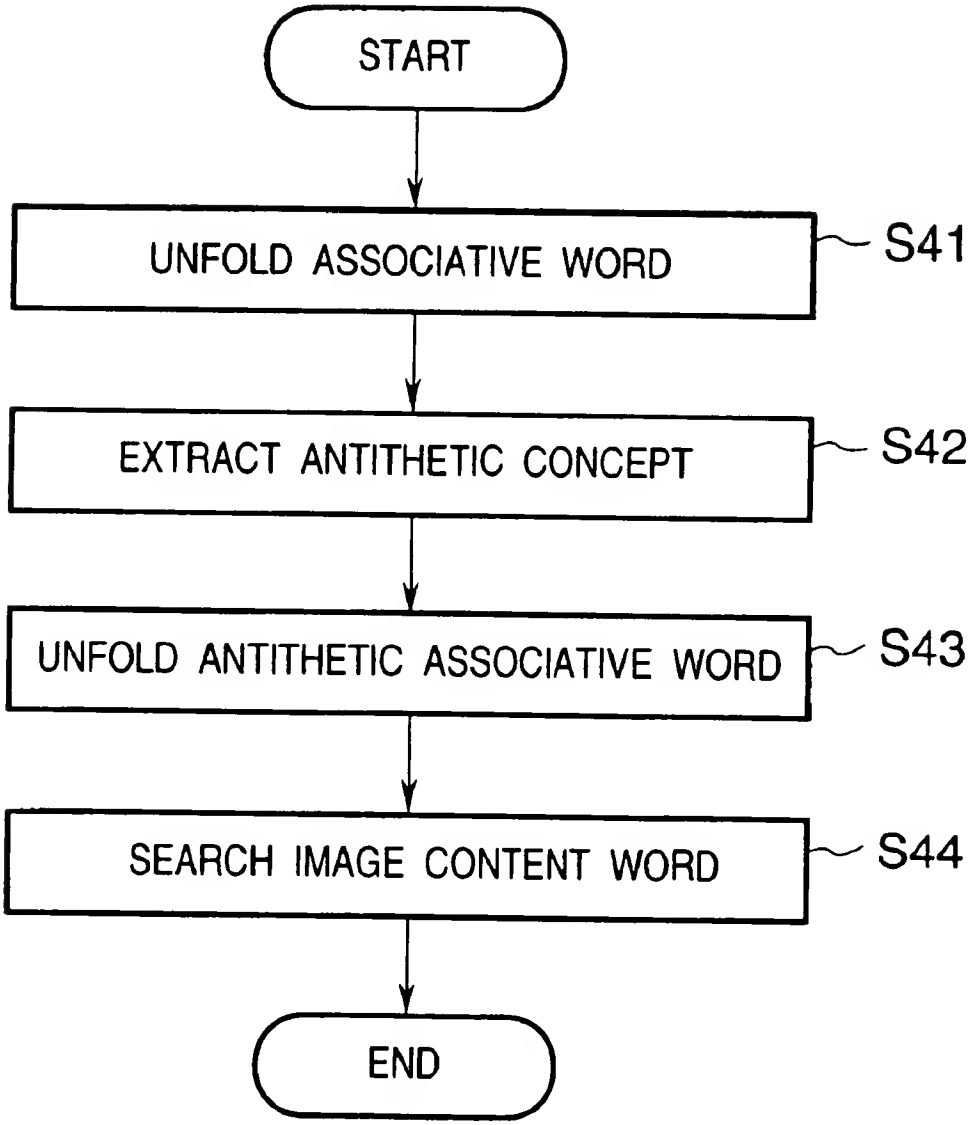

START
UNFOLD ASSOCIATIVE WORD
S41
EXTRACT ANTITHETIC CONCEPT
S42
UNFOLD ANTITHETIC ASSOCIATIVE WORD
S43
SEARCH IMAGE CONTENT WORD
S44
END

FIG. 21

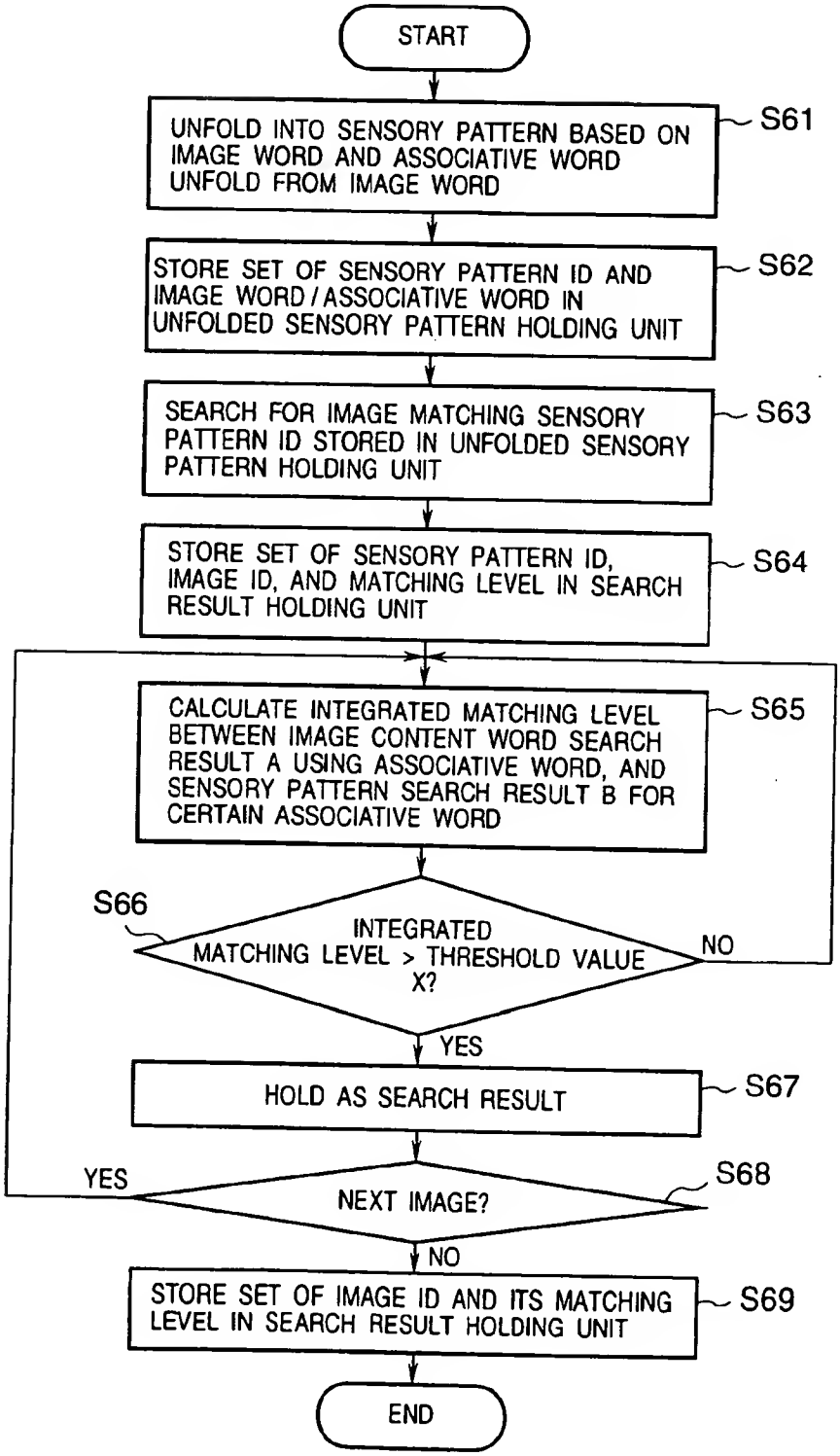

START
UNFOLD INTO SENSORY PATTERN BASED ON IMAGE WORD AND ASSOCIATIVE WORD UNFOLD FROM IMAGE WORD
S61
STORE SET OF SENSORY PATTERN ID AND IMAGE WORD / ASSOCIATIVE WORD IN UNFOLDED SENSORY PATTERN HOLDING UNIT
S62
SEARCH FOR IMAGE MATCHING SENSORY PATTERN ID STORED IN UNFOLDED SENSORY PATTERN HOLDING UNIT
S63
STORE SET OF SENSORY PATTERN ID, IMAGE ID, AND MATCHING LEVEL IN SEARCH RESULT HOLDING UNIT
S64
CALCULATE INTEGRATED MATCHING LEVEL BETWEEN IMAGE CONTENT WORD SEARCH RESULT A USING ASSOCIATIVE WORD, AND SENSORY PATTERN SEARCH RESULT B FOR CERTAIN ASSOCIATIVE WORD
S65
S66
INTEGRATED MATCHING LEVEL > THRESHOLD VALUE X?
NO
YES
HOLD AS SEARCH RESULT
S67
YES
NEXT IMAGE?
S68
NO
STORE SET OF IMAGE ID AND ITS MATCHING LEVEL IN SEARCH RESULT HOLDING UNIT
S69
END

FIG. 22

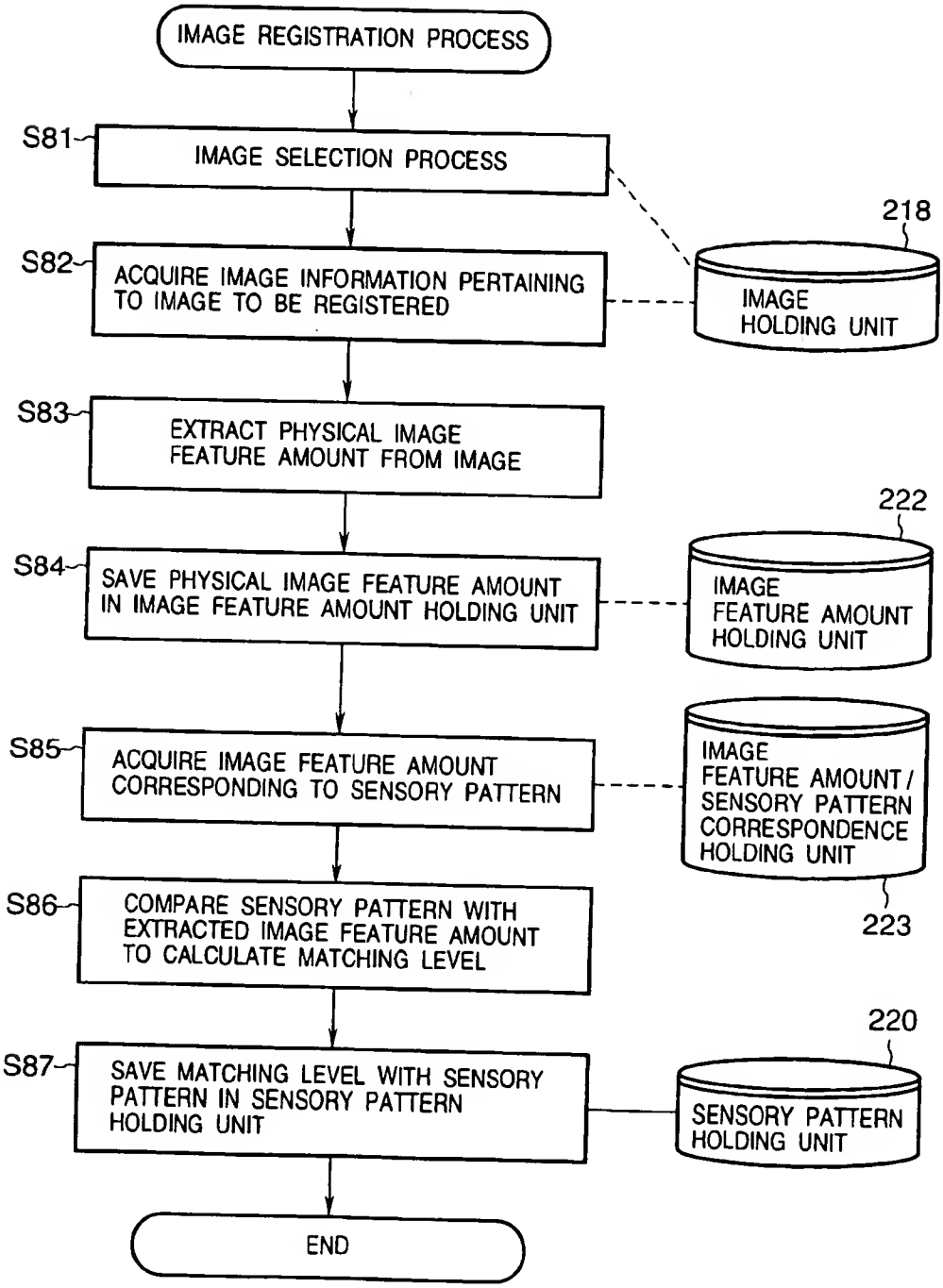


IMAGE REGISTRATION PROCESS
S81
IMAGE SELECTION PROCESS
S82
ACQUIRE IMAGE INFORMATION PERTAINING TO IMAGE TO BE REGISTERED
218
IMAGE HOLDING UNIT
S83
EXTRACT PHYSICAL IMAGE FEATURE AMOUNT FROM IMAGE
222
S84
SAVE PHYSICAL IMAGE FEATURE AMOUNT IN IMAGE FEATURE AMOUNT HOLDING UNIT
IMAGE FEATURE AMOUNT HOLDING UNIT
S85
ACQUIRE IMAGE FEATURE AMOUNT CORRESPONDING TO SENSORY PATTERN
IMAGE FEATURE AMOUNT / SENSORY PATTERN CORRESPONDENCE HOLDING UNIT
223
S86
COMPARE SENSORY PATTERN WITH EXTRACTED IMAGE FEATURE AMOUNT TO CALCULATE MATCHING LEVEL
220
S87
SAVE MATCHING LEVEL WITH SENSORY PATTERN IN SENSORY PATTERN HOLDING UNIT
SENSORY PATTERN HOLDING UNIT
END

INFORMATION SEARCH APPARATUS AND METHOD

BACKGROUND OF THE INVENTION

The present invention relates to an information search apparatus and method for searching information on the basis of an input query word. More specifically, the present invention relates to an information search apparatus and method for managing a plurality of kinds of multimedia information, and searching the managed multimedia information for desired multimedia information, and a computer readable memory.

A conventional information search apparatus, which searches multimedia information, e.g., image information, makes a search using data (keywords) derived from subjective evaluation results of one or a plurality of persons for test images, pbysical image features extracted from images, and the like.

When an image is searched for using a keyword, a required image is obtained by matching a given keyword with that corresponding to the image. Also, a scheme for obtaining an image, that cannot be obtained by full-word matching with an input keyword, by matching not only the input keyword but also an associated keyword associated with the input keyword with a keyword corresponding to an image, is proposed. Furthermore, a search scheme wbich obtains an image with similar color information by detecting a correspondence between the input keyword and color information using, e.g., color information of images is proposed.

In the image search using keywords, an impression that a person receives upon watching an image, or key information linked with the impression is appended to image information and is used in search. As the key information, words that express impressions evoked by images such as "warm", "cold", and the like, and words tbat represent objects in drawn images such as "kitty", "sea", "mountain", and the like are appended as keywords. Also, local image feature components on drawn images are subjectively evaluated and are often appended as key information. For example, information that pertains to a color such as "red", "blue", and the like, information tbat pertains to a shape sucb as "round", "triangular", "sharp", and the like, and information that pertains to a texture such as "sandy", "smooth", and the like are expressed using words and icons, are appended to images as key information, and are used in search.

In a system in which physical image feature amounts are extracted from images, and are used in image search, image features include local colors painted on images, overall color tones, and shapes, compositions, textures, and the like of objects on drawn images. An image feature amount is extracted from segmented regions or blocks obtained by segmenting the overall image into regions based on color information, or segmenting the image into blocks eacb having a given area, or is extracted from the entire image. Physical image features include, e.g., color information, density distribution, texture, edge, region, area, position, frequency distribution, and the like of an image.

However, in the above searcb scheme, wben an image including a keyword that matches the input query word is searcbed for, images wbich do not match the search request of the searcher are often obtained. Especially, when an image search is made using an abstract query word such as a "refreshing" image, images found by the search are limited. To solve this problem, a search may be made by

unfolding the query word "refreshing" to keywords which are associated with thai query word. However, when such scbeme is used, images which are not "refresbing" may be mixed in search results.

In this manner, the operator cannot designate query conditions for obtaining a desired search result with respect to a search request indicated by the input keyword, and cannot obtain an intended search result. For example, even when the operator wants to find only images having "refreshing" feature patterns with respect to a search request "refreshing", images having content words associated from the search request "refreshing" such as a music score of a "refreshing" music, a "refreshing" athlete", and the like, are presented, i.e., images which do not match the search request are presented.

In place of a query word, a query image may be input, and a search may be made using its feature amount. However, in this case, a query image which reflects the searcher's will must be prepared, and it is difficult to select a query image, resulting in poor operability.

SUMMARY OF THE INVENTION

The present invention has been made in consideration of the above-mentioned problems, and has as its object to provide an image search method and apparatus which can extract the information wanted with high precision with respect to an input query word.

In order to achieve the above object, according to one aspect of the present invention, there is provided an information search apparatus for searching information based on an input query word, comprising first search means for determining a query keyword on the basis of the query word, and searching information on the basis of the query keyword, second search means for determining a feature amount corresponding to the query word, and searching information on the basis of the feature amount, setting means for setting a search weight to be assigned to searcb results of the first and second search means, and integration means for integrating search results obtained by the first and second search means in accordance with the search weight set by the setting means.

In order to achieve the above object, according to another aspect of the present invention, there is provided an information search method for searching information based on an input query word, comprising the first searcb step of determining a query keyword on the basis of the query word, and searching information on the basis of the query keyword, the second search step of determining a feature amount corresponding to the query word, and searching information on the basis of the feature amount, the setting step of setting a searcb weight to be assigned to search results in tbe first and second search steps, and the integration step of integrating search results obtained in the first and second search steps in accordance with the search weight set in the setting step.

Other features and advantages of the present invention will be apparent from the following description taken in conjunction with the accompanying drawings, in which like reference characters designate the same or similar parts throughout the figures thereof.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of the specification, illustrate embodiments of the invention and, together with the description, serve to explain the principles of the invention

FIG. 1 is a block diagram showing an example of the arrangement of an information processing apparatus which constructs an image search apparatus according to an embodiment of the present invention;

FIG. 2 is a block diagram depicting the processing arrangement in the image search apparatus according to the embodiment of the present invention;

FIG. 3 is a view showing a display example of search perspectives in association with a search request word input in a search request input process 201;

FIG. 4 is a view showing a display example of a weight designation control panel for designating a search weight for a search using associative words, and a search weight for a search using sensory patterns, in the search request word input in a search request input process 201;

FIG. 5 is a table showing the data structure of an image holding unit 218 which stores image IDs in correspondence with image file storage paths;

FIG. 6 is a table showing an example of the data structure of an image content word holding unit 219 which stores image IDs in correspondence with image content words;

FIG. 7 is a table which stores data of the image content word holding unit shown in FIG. 6 as a list of image IDs using image content words as keys;

FIG. 8 is a table showing an example of the data structure of a concept discrimination dictionary 205;

FIG. 9 is a table showing an example of the data structure of an associative word dictionary 211;

FIG. 10 is a table for explaining the data holding format in a search result holding unit 216;

FIG. 11 is a table for explaining an example of the data structure of an unfolded sensory pattern holding unit 213 shown in FIG. 2;

FIG. 12 is a table showing an example of the data structure of an image word/sensory pattern correspondence holding unit 215 shown in FIG. 2;

FIG. 13 is a table showing the data structure of a sensory pattern holding unit 220 shown in FIG. 2;

FIG. 14 is a table showing a data example obtained upon extracting image feature amounts from a single image by an image feature amount extraction process;

FIG. 15 is a table showing an example of image feature amounts in this embodiment, which are obtained by extracting representative colors in units of image regions/blocks;

FIG. 16 is a table showing a storage example of an image feature amount holding unit 222 shown in FIG. 2;

FIG. 17 is a table showing a data storage example of an image feature amount/sensory pattern correspondence holding unit 223 shown in FIG. 2;

FIG. 18 is a flow chart for explaining the operation of the present invention;

FIG. 19 is a flow chart showing the details of the search request input process 201 (step S1 in FIG. 18);

FIG. 20 is a flow chart showing the details of an associative word unfolding process 208 and an image content word search process 210 using associative words (step S4 in FIG. 18);

FIG. 21 is a flow chart showing the details of a sensory pattern unfolding process 212, sensory pattern search process 214, and search result integration process 217; and

FIG. 22 is a flow chart showing an example of a preprocess of a search, which is done upon registering an image.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

The preferred embodiments of the present invention will now be described in detail in accordance with the accompanying drawings.

FIG. 1 shows an example of the arrangement of an information processing apparatus which constitutes an image search apparatus according to this embodiment.

Referring to FIG. 1, reference numeral 11 denotes a microprocessor (to be referred to as a CPU hereinafter), which makes computations, logical decisions, and the like for image information search in accordance with control programs, and controls individual building components connected to an address bus AB, control bus CB, and data bus DB via these buses. The address bus AB transfers an address signal indicating the building component to be controlled by the CPU 11. The control bus CB transfers a control signal for each building component to be controlled by the CPU 11. The data bus DB transfers data among the respective building components.

Reference numeral 12 denotes a read-only memory (to be referred to as a ROM hereinafter), which stores a boot processing program and the like executed by the CPU 11 upon starting up the apparatus of this embodiment. Reference numeral 13 denotes a rewritable random access memory (to be referred to as a RAM hereinafter) which is configured by 16 bits per word, and is used as a temporary storage of various data from the respective building components. Especially, in this embodiment, the RAM 13 provides various data holding units such as a query word holding unit 202, search perspective holding unit 203, search weight holding unit 204, determined weight holding unit 207, unfolded associative word holding unit 209, unfolded sensory pattern holding unit 213, and search result holding unit 216, which will be described later with reference to FIG. 2.

Reference numeral 14 denotes an external memory (DISK), which stores a concept discrimination dictionary 205 and associative word dictionary 211, and provides data holding units such as an image word/sensory pattern correspondence holding unit 215, image content word holding unit 219, image holding unit 218, sensory pattern holding unit 220, image feature amount holding unit 222, and image feature amount/sensory pattern correspondence holding unit 223, which will be described later with reference to FIG. 2. As a storage medium of the external memory 14, a ROM, floppy disk, CD-ROM, memory card, magnetooptical disk, or the like can be used.

Also, the external memory 14 stores programs for respectively implementing the respective processes, i.e., a search request input process 201, weight determination process 206, associative word unfolding process 208, image content word search unit 210 using associative words, sensory pattern unfolding process 212, sensory pattern search process 214, search result integration process 217, image feature amount extraction process 221, and sensory pattern determination process 224, which will be described later with reference to FIG. 2 and the like. These programs are loaded onto the RAM 13 as needed, and are executed by the CPU 11.

Reference numeral 15 denotes a keyboard (KB) which has alphabet keys, hiragana keys, katakana keys, character symbol input keys for inputting a period, comma, and the like, a search key for instructing a search (a function key on a general keyboard may be used instead), and various function keys such as cursor moving keys for instructing cursor

movement, and the like. Also, a pointing device such as a mouse or the like (not shown) may be connected.

Reference numeral **16** denotes a display video memory (to be referred to as VRAM hereinafter) for storing a pattern of data to be displayed. Reference numeral **17** denotes a CRT controller (to be referred to as a CRTC hereinafter) for displaying the contents stored in the VRAM **16** on a CRT **18**. Reference numeral **18** denotes a display device (CRT) using, e.g., a cathode ray tube, or the like. The dot display pattern and cursor display on the CRT **18** are controlled by the CRTC **17**. Note that various other displays such as a liquid crystal display, and the like may be used as the display device. Reference numeral **19** denotes a network controller (NIC), which connects the image search apparatus of this embodiment to a network such as Ethernet or the like.

The image search apparatus of this embodiment constituted by the aforementioned building components operates in accordance with various inputs from the keyboard **15** an various inputs supplied from the network controller via the network. Upon receiving the input from the keyboard **15** or network controller **19**, and interrupt signal is set to the CPU **11**. Upon receiving the interrupt signal, the CPU **11** reads out various control data stored in the external memory **14**, and executes various kinds of control in accordance with these control data. Also, the present invention is achieved by supplying a storage medium that stores a program according to the present invention to a system or apparatus, and by reading out and executing program codes stored in the storage medium by a computer of the system or apparatus.

FIG. **2** is a block diagram depicting the processing arrangement in the image search apparatus of this embodiment.

Referring to FIG. **2**, reference numeral **201** denotes a search request input process for inputting query items (query word, search perspective or category, and search weight in this embodiment; to be described in detail later) that pertain to the information wanted. Reference numeral **202** denotes a query word holding unit for storing a query word input by the search request input process **201**. Reference numeral **203** denotes a search perspective holding unit for storing a search perspective input by the search request input process **201**. Reference numeral **204** denotes a search weight holding unit for storing a search weight input by the search request input process **201**.

Reference numeral **205** denotes a concept discrimination dictionary having a search perspective that pertains to a concept as information wanted, an antithetic concept having a meaning contrary or antonymous to the concept as the information wanted, and two kinds of coefficients (associated weight and sensory pattern weight) for weight discrimination upon searching for a concept, as shown in FIG. **8**. Note that the concept discrimination dictionary **205** will be described in detail later with reference to FIG. **8**. Reference numeral **206** denotes a weight determination process for giving weights (associated weight and sensory pattern weight) indicating the weight balance on associative words (obtained by an associative word unfolding processing unit **208**) and sensory patterns (obtained by a sensory pattern unfolding processing unit **212**) in actual search using a query word stored in the query word holding unit **202**. Reference numeral **207** denotes a determined weight holding unit for holding the search weight determined by the weight determination process **206**.

Reference numeral **208** denotes an associative word unfolding process for unfolding the query word stored in the query word holding unit **202** into associative words with reference to an associative word dictionary **211**, obtaining an antithetic concept antonymous to that query word from the concept discrimination dictionary **205**, and unfolding the obtained antithetic concept into associative words with reference to the associative word dictionary **211**. Reference numeral **209** denotes an unfolded associative word holding unit for holding the associative words (including those of the antithetic concept) unfolded by the associative word unfolding process **208**. Reference numeral **210** denotes an image content word search process using associative words (to be referred to as an image content word search process hereinafter), which finds image content words, that are stored in an image content word holding unit **219** and match the unfolded associative words, by search with reference to the unfolded associative word holding unit **209**. Reference numeral **211** denotes an associative word dictionary for storing associative words in units of concepts serving as index words in correspondence with associative perspectives (this process will be described in more detail later with reference to FIG. **9**).

Reference numeral **212** denotes a sensory pattern unfolding process for unfolding the query word stored in the query word holding unit **202** into sensory patterns with reference to an image word/sensory pattern correspondence holding unit **215**, obtaining an antithetic concept antonymous to that query word from the concept discrimination dictionary **205**, and unfolding the obtained antithetic concept into sensory patterns with reference to the image word/sensory pattern correspondence holding unit **215**.

Reference numeral **213** denotes an unfolded sensory pattern holding unit for temporarily storing the sensory patterns unfolded by the sensory pattern unfolding process **212**. Storage of data in the unfolded sensory pattern holding unit **213** will be described later with reference to FIG. **11**. Reference numeral **214** denotes a sensory pattern search process for finding sensory patterns, which are stored in the sensory pattern holding unit **220** and are similar to the unfolded sensory patterns, by search with reference to the sensory pattern holding unit **220**.

Reference numeral **215** denotes an image word/sensory pattern correspondence holding unit for storing the correspondence between the image words and sensory patterns by storing sensory pattern IDs corresponding to sets of image words and associative words associated with the image words. Note that the image word/sensory pattern correspondence holding unit **215** will be described in detail later with reference to FIG. **12**.

Reference numeral **216** denotes a search result holding unit for storing image IDs found by searches of the image content word search process **210** and sensory pattern search process **214**. Reference numeral **217** denotes a search result integration process for integrating the search results of image content words using the associative words, and the search results of sensory patterns stored in the search result holding unit **216**, on the basis of the search weights obtained by the weight determination process **206** and stored in the determined weight holding unit **207**.

Reference numeral **218** denotes an image holding unit for storing image information serving as test images. Reference numeral **219** denotes an image content word holding unit for verbalizing and storing concepts appended to images stored in the image holding unit **218** to express their contents. Reference numeral **220** denotes a sensory pattern holding unit for storing matching levels between image information of the individual images stored in the image holding unit **218** and sensory patterns. More specifically, the sensory pattern

holding unit **220** stores the matching levels with sensory patterns in units of image IDs. The data contents of the sensory pattern holding unit **220** will be described later with reference to FIG. **13**.

Reference numeral **221** denotes an image feature extraction process for extracting physical image feature amounts from image information stored in the image holding unit **218**. Physical image feature amounts are visual features or signatures extracted from regions segmented on the basis of color information, blocks each segmented to have a given area, or the entire image. For example, the image feature is numerical information such as the color distribution or histogram, density distribution, texture, edge, frequency distribution, and the like of an image. Note that the image feature amounts will be described in detail later with reference to FIG. **14**.

Reference numeral **222** denotes an image feature amount holding unit for storing the image feature amounts obtained by the image feature amount extraction process **221**. Reference numeral **223** denotes an image feature amount/sensory pattern correspondence holding unit for storing image feature amounts and sensory patterns in correspondence with each other. The image feature amount/sensory pattern corresponding holding unit **223** stores sensory pattern IDs and image feature amount data corresponding to those IDs. Note that the data structure of the image feature amount/sensory pattern correspondence holding unit **223** will be described in detail later with reference to FIG. **17**.

Reference numeral **224** denotes a sensory pattern determination process for comparing image feature amounts extracted from a sensory pattern and image information and determining their matching level with reference to the image feature amount holding unit **222** and image feature amount/sensory pattern correspondence holding unit **223**, and registering the result in the aforementioned sensory pattern holding unit **220**.

A display example of a search perspective that pertains to search request items input at the search request input processing unit **201** will be explained below with reference to FIG. **3**.

FIG. **3** shows a display example of search perspectives in association with the query word input by the search request input process **201**. When a query word is input by operating, e.g., the keyboard **15**, the concept discrimination dictionary **205** shown in FIG. **8** is searched using the query word as an index word to extract corresponding search perspectives. FIG. **3** illustrates that three search perspectives "color tone", "taste", and "general atmosphere" are available in relation to a query word "mild", and hatched "color tone" is selected as the search perspective. When the user presses (clicks) an OK button in this state, the search perspective "color tone" is selected, and is held in the search perspective holding unit **203**.

By pressing one of the cursor moving keys on the keyboard **15**, the hatching moves from "color tone" to "taste" or "general atmosphere", and the user can designate a desired search perspective or category. Note that "mild" as the query word is held in the query word holding unit **202**, and the selected search perspective ("color tone" in FIG. **3**) is held in the search perspective holding unit **203**.

A display example on the control panel when the operator instructs the search weight balance on a search using associative words and a search using sensory patterns in actual search will be explained below with reference to FIG. **4**.

FIG. **4** shows a display example of a weight designation control panel for instructing search weights for searches using associative words and sensory patterns in the search request input process **201**. As described above, in this embodiment, a search using associative words and a search using the feature amounts of images (sensory patterns) based on the query word are made, and the search results are integrated. In this integration process, the two search results are weighted. On the weight designation control panel, the user can designate a search weight for a search using associative words, and that for a search using sensory patterns. That is, the user can designate the weight balance on a search using associative words or that using sensory patterns in actual search.

Referring to FIG. **4**, when the user slides a slide button **41** to the left, an instruction that sets a heavier weight on a search using associative words is issued; when he or she slides the slide button **41** to the right, an instruction that sets a heavier weight on a search using sensory patterns is issued. When the user designates search weights using the slide button **41** and then presses an OK button **43**, a search weight instruction is issued. Note that a button **42** is pressed when no, search weights are clearly designated. Upon depression of the button **42**, predetermined weight values (which are obtained from an associated weight **83** and sensory pattern weight **84** in the concept discrimination dictionary **205** (FIG. **8**)) are used. The set weights are stored in the search weight holding unit **204**. Note that the buttons **41** to **43** on the control panel may be clicked by a pointing device (not shown).

The structure of the image holding unit **218** will be described below using FIG. **5**.

FIG. **5** shows the data structure of the image holding unit **218** which stores image IDs in correspondence with image file storage paths. Referring to FIG. **5**, reference numeral **51** denotes an image ID which is an identification number uniquely assigned to one image file in this image database. Reference numeral **52** denotes a file path which indicates the storage location of an image file corresponding to the image ID in the external memory **14**, and corresponds to the directory and file of MS-DOS.

An image file is divided into header and image data fields (not shown in FIG. **5**). The header field stores information required for reading data from that image file, and additional information that explains the image contents. As such information, an image format identifier indicating the image format name of the image, file size, image width, height, and depth, the presence/absence of compression, color pallet information, resolution, offset to the storage location of image data, and the like are stored. The image data field stores image data in turn. This embodiment uses the BMP format of Microsoft Corp. as such image format, but other compression formats such as GIF, JPEG, FlashPix, and the like may be used.

The structure of the image content word holding unit **219** will be described below with the aid of FIG. **6**.

FIG. **6** shows an example of the data structure of the image content word holding unit **219** which stores image IDs in correspondence with image content words. Referring to FIG. **6**, reference numeral **61** denotes an image ID, which corresponds to the image ID **51** in FIG. **5**. Reference numeral **62** denotes an image content word which stores a word for expressing each image identified by the image ID **61**. The image content word is a keyword which verbally expresses an image feature in an image, and is stored as a character code (e.g., unicode). A plurality of keywords may be stored per image, and the image content word holding unit **219** is constructed as a list of image content words **62** using image IDs **61** as keys.

FIG. 7 shows a table which stores the data of the image content word holding unit shown in FIG. 6 as a list of image IDs using image content words as keys. Referring to FIG. 7, all image IDs that contain image content words 71 as keywords are stored as image IDs 72.

The structure of the concept discrimination dictionary 205 will be described below using FIG. 8.

FIG. 8 shows an example of the data structure of the concept discrimination dictionary 205. The concept discrimination dictionary 205 provides information that pertains to a query word serving as a search request, and is a table which stores index words 80 corresponding to query words, search perspectives 81 associated with index words, antithetic concepts 82 having meanings contrary to the index words in units of search perspectives, associative word weights 83 used upon searching the index words, and sensory pattern weights 84 used upon searching the index words in correspondence with each other.

The structure of the associative word dictionary 211 will be explained below with reference to FIG. 9.

FIG. 9 shows an example of the data structure of the associative word dictionary 211. The associative word dictionary 211 is composed of associative IDs 90 each of which assigns a unique number to a set of associative words for each index word, index words 91 each serving as a start point of association, associative words 92 evoked by the index words 91, associative perspectives 93 which are relevant to associations of the associative words 92, and association strengths 94 each indicating the strength of association between each pair of index word 91 and associative word 92.

In this embodiment, the association strength 94 assumes an absolute value ranging from 0 to 10, and its sign indicates direction of association of the associative word. More specifically, when the association strength is a positive value, it indicates a stronger associative relationship (higher bilateral association) as the association strength value is larger; when the association strength is a negative value, it indicates a harder associative relationship as the association strength value is larger. For example, an associative word "folkcraft article" corresponding to an index word "simple" in associative data with the associative ID=126533 can be associated with strength "6", but an associative word "chandelier" in associative data with the associative ID=126536 is hardly associated with strength "9" since its association strength is a negative value.

The structure of the search result holding unit 216 will be described below with reference to FIG. 10.

FIG. 10 shows the data holding format in the search result holding unit 216. As described above, the search result holding unit 216 stores image IDs which are found by searches of the image content word search process 210 using associative words and the sensory pattern search process 214.

Referring to FIG. 10, reference numeral 100 denotes a field for storing image IDs found by search; 101, a field for storing the number of matched associative words with positive association strengths by the image content word search process 217. An associative word ID list 102 stores a maximum of 20 associative word IDs 90 (FIG. 9) of matched associative words in the associative word dictionary 211. When the number 101 of matched associative words is zero, the associative ID 102 is filled with NULL code. Reference numeral 103 denotes a field for storing associative matching levels of associative words with respect to the image IDs 100. When the number 101 of matched associative words is zero, the association matching level 103 also stores zero.

Reference numeral 104 denotes a field for storing the number of sensory patterns with highest similarity, which are found by search by the sensory pattern search process 223. Reference numeral 105 denotes a sensory pattern ID list which stores a maximum of 20 sensory pattern IDs of matched sensory patterns. When the number 104 of matched sensory patterns is zero, the sensory pattern ID 105 is filled with NULL code. Reference numeral 106 denotes a field for storing the search matching level of a sensory pattern search with respect to the image ID 100. When the number 104 of matched sensory patterns is zero, the sensory pattern matching level 106 stores zero. Reference numeral 107 denotes a field for storing an integrated matching level (obtained by the search result integration process 217) of the image ID 100 with respect to the search request, which is calculated using the associative matching level 103 and sensory pattern matching level 106 as parameters.

The structure of the above-mentioned unfolded sensory pattern holding unit 213 will be described in detail below with reference to FIG. 11.

FIG. 11 is a table for explaining an example of the data structure of the unfolded sensory pattern holding unit 213 shown in FIG. 2. Referring to FIG. 11, reference numeral 110 denotes an image word as an unfolding source from which this sensory pattern has evolved upon unfolding, and the same image word as that in the query word holding unit 202 is stored. In this embodiment, a character string "refreshing" is stored, and ends with NULL code. Reference numeral 111 denotes the number of sensory patterns obtained by unfolding the image word 110 with reference to the image word/sensory pattern correspondence holding unit 215. For example, when the contents of the image word/sensory pattern correspondence holding unit 215 are as shown in FIG. 12, the number of sensory patterns unfolded from the image word "refreshing" is 7. Reference numeral 112 denotes an address indicating the storage location area of data obtained by actually unfolding the image word "refreshing". In the example shown in FIG. 11, the storage location address 112 is linked with unfolded data 115 shown in FIG. 11.

In the data 115, data actually unfolded from "refreshing", i.e., sets of associative words and sensory patterns corresponding to the number 111 of sensory patterns are stored. In this embodiment, seven sets of associative words and sensory patterns are stored. For example, an associative word 114 is that of the image word "refreshing" and stores a character string "forest" in this example. Also, a sensory pattern ID 113 corresponds to the image word "refreshing" and its associative word "forest", and stores "5" in this example. The same applies to other sets of associative words and sensory patterns.

The structure of the aforementioned image word/sensory pattern correspondence holding unit 215 will be described in detail below using FIG. 12.

FIG. 12 shows an example of the data structure of the image word/sensory pattern correspondence holding unit 215 in FIG. 2. Referring to FIG. 12, reference numeral 120 denotes an image word serving as an unfolding source of this sensory pattern. In FIG. 12, character strings "refreshing", "tropical", and the like are stored, and end with NULL code. Reference numeral 121 denotes an associative word unfolded from each image word 120. In this embodiment, associative words "forest", "tableland", "blue sky", and the like are stored in correspondence with "refreshing", and

these character strings end with NULL code. When no character string is stored in this field **121**, i.e., NULL code alone is stored, this sensory pattern applies to all image words "refreshing"; no specific associative word has been designated.

Reference numeral **122** denotes a sensory pattern ID corresponding to the image word **120** and associative word **121**. In this embodiment, "005" and "006" are stored as sensory pattern IDs corresponding to the image word "refreshing" and its associative word "forest". Also, sensory patterns for "not refreshing" as an antithetic concept of "refreshing" are stored. In this embodiment, for "not refreshing", no associative words are registered and "001" and "010" are registered as sensory pattern IDs.

The structure of the above-mentioned sensory pattern holding unit **220** will be described in detail below using FIG. **13**.

FIG. **13** shows the data structure of the sensory pattern holding unit **220** in FIG. **2**. Referring to FIG. **13**, reference numeral **131** denotes an image ID for identifying an image to be registered. The image IDs **131** use the same ones as those stored in the image holding unit **218**, and uniquely define images in this system. A field **132** stores sensory pattern IDs. In this embodiment, since the matching levels between each image and all sensory patterns stored in the image feature amount/sensory pattern correspondence holding unit **223** are calculated, all the sensory pattern IDs (1 to m) are stored. Reference numeral **133** denotes a numerical value indicating the matching level between each image and sensory pattern. The matching level assumes a value ranging from 0 to 1; 0 indicates the image does not match the sensory pattern at all, and the matching level becomes higher as it is closer to 1. For example, in this embodiment, the matching level between image with the image ID=001 and sensory pattern **1** is 0.10, and the matching level between that image and sensory pattern **2** is 0.

The aforementioned image feature amounts will be explained in detail below with reference to FIG. **14**.

FIG. **14** shows a data example obtained upon extracting the image feature amounts from one image by the image feature amount extraction process. In FIG. **14**, X1, X2, X3, . . . , Xn represent image features, B1, B2, . . . , Bm represent regions/blocks from which image feature amounts are extracted, and x11 to xmn represent image feature amounts extracted from the individual regions/blocks. That is, feature amounts that pertain to physical image features X1 to Xn are obtained in units of regions/blocks.

FIG. **15** exemplifies a case wherein chromatic image feature amounts are especially extracted. In this case, representative colors are extracted in units of regions or blocks of an image. Referring to FIG. **15**, a physical image feature is expressed by "representative color" and "feature amount", representative colors extracted from regions/blocks B1, B2, . . . , Bn are C1(R1, G1, B1), C2(R2, G2, B2), . . . , Cn(Rn, Gn, Bn), and their image feature amounts are c1 to cn.

The structure of the image feature amount holding unit **222** will be described below using FIG. **16**.

FIG. **16** shows a storage example of the image feature amount holding unit **222** in FIG. **2**. Referring to FIG. **16**, reference numeral **161** denotes an image ID for identifying an image to be registered. The image IDs **161** use the same ones as those stored in the image holding unit **218**. Reference numeral **162** denotes a block or region number from which an image feature amount is extracted. In this embodiment, B1, B2, . . . , Bm represent the region/block numbers. Reference numeral **163** denotes information (in this embodiment, a representative color is used) indicating an image feature extracted from each region or block (B1, B2, . . . , Bm). This embodiment exemplifies a case wherein chromatic image features are extracted, and for example, a plurality of pieces of information C11(R11, G11, B11), . . . , Cn1(Rn1, Gn1, Bn1) indicating colors are stored. Reference numeral **164** denotes image feature amounts of image features extracted from the individual regions/blocks. In this embodiment, c11, . . . , cn1 are stored as the feature amounts of C11(R11, G11, B11), . . . , Cn1(Rn1, Gn1, Bn1).

The structure of the image feature amount/sensory pattern correspondence holding unit **223** will be described in detail below using FIG. **17**.

FIG. **17** shows a data storage example of the image feature amount/sensory pattern correspondence holding unit **223** in FIG. **2**. Referring to FIG. **17**, reference numeral **171** denotes a sensory pattern ID, which uniquely identifies a sensory pattern. Reference numeral **172** denotes image feature amount data corresponding to each sensory pattern ID. In this embodiment, a sensory pattern is expressed by a chromatic image feature amount, and a combination of color components (values in a color space such as RGB, HVC, or the like) corresponding to each sensory pattern ID is stored. In this embodiment, RGB values assume integers ranging from 0 to 255. A maximum of m colors correspond to each sensory pattern ID.

The sensory pattern determination process **224** calculates the matching levels between each of image data registered in the image holding unit **218** and the respective sensory patterns using the aforementioned image feature amount holding unit **222** and image feature amount/sensory pattern correspondence holding unit **223**, and registers them in the sensory pattern holding unit **220** (to be described later in step S87 in FIG. **22**).

The processes executed in the image search apparatus of this embodiment with the above arrangement will be described below.

FIG. **18** is a flow chart showing the operation of the image search apparatus of this embodiment. Referring to FIG. **18**, step S1 is a processing module that implements the search request input process **201** in FIG. **2**, and inputs a search result. Note that the details of this process will be explained in detail later with reference to FIG. **19**.

Step S2 is a processing module that implements the weight discrimination process **206**, and if it is determined with reference to the contents stored in the search weight holding unit **204** in the search request input process **201** in step S1 that search weights are designated, the designated values are stored in the determined weight holding unit **207**. On the other hand, if no search weights are designated, index words **80** are searched for a query word stored in the query word holding unit **202** with reference to the concept discrimination dictionary **205** (FIG. **8**) so as to read out a corresponding associated weight **83** and sensory pattern weight **84**, and the readout weights are stored in the determined weight holding unit **207**. If there is no index word **80** that is relevant to the contents of the query word holding unit **202**, a default value "5" is stored as both the associated and sensory pattern weights in the determined weight holding unit **207**.

It is checked with reference to the determined weight holding unit **207** in step S3 if the associated weight is zero. If the associated weight is zero, the flow advances to step S5; otherwise, the process in step S4 is executed. Step S4 is a processing module that implements the associative word unfolding process **208** and image content word search

process **210** using associative words in FIG. **2**, and this process will be described in detail later with reference to FIG. **20**.

It is checked with reference to the determined weight holding unit **207** in step S5 if the sensory pattern weight is zero. If the sensory pattern weight is zero, since a search using sensory pattern is unnecessary, the flow advances to step S7; otherwise, the process in step S6 is executed. Step S6 is a processing module that implements the sensory pattern unfolding process **212** and sensory pattern search process **214** in FIG. **2**, and will be described in detail later with reference to FIG. **21**.

Step S7 is a processing module that implements the search result integration process **217** in FIG. **2**, and will be described in detail later with reference to FIG. **22**.

In step S8, image files corresponding to image IDs stored in the search result holding unit **216** as search results obtained in step S7 are read out from the image holding unit **218**, and are displayed. Note that the search result display process in step S8 is a known one which is prevalent in image search apparatuses of the same type.

FIG. **19** is a flow chart showing the details of the search request input process **201** (step S1 in FIG. **18**). In step S21, a query word serving as a search request is input. The query word input is attained by storing a character code input at the keyboard **15** in the query word holding unit **202** on the RAM **13**.

In step S22, using the query word stored in the query word holding unit **202** as a search request, search perspectives that are relevant to the search request are extracted from the concept discrimination dictionary **205**. In this case, all search perspectives corresponding to index words **80** (FIG. **8**), which match the query word in the query word holding unit **202**, are extracted. For example, when the query word is "mild", three search perspectives "color tone", "taste", and "general atmosphere" can be obtained.

It is checked in step S23 if a search perspective or perspectives is or are found by search perspective extraction in step S22. If a search perspective or perspectives is or are found, the flow advances to step S24; otherwise, the flow advances to step S26.

If search perspectives are found in step S22, they are displayed together with the query word, as in shown in FIG. **3**, in step S24. In step S25, the user selects a desired one of the displayed search perspectives using the user interface that has been described previously with reference to FIG. **3**. The selected search perspective is stored in the search perspective holding unit **203**.

In step S26, the user inputs search weights which determine the weight balance on a search using associative words and a search using sensory pattern in actual search in relation to the search process method in response to the search request. In this embodiment, the user sets the weights using the user interface shown in FIG. **4**. That is, the user operates the slide bar shown in FIG. **4** to designate the weight ratios (desired search weights) on associative words and sensory patterns by the length of the horizontal bar of the slider (the position of the button **41**). When the user does not designate any search weights, he or she designates use of the default values of the weights shown in FIG. **4**.

It is checked in step S27 if search weights are designated. If it is instructed to use the default weight values, the processing ends. On the other hand, if search weights are designated, the flow advances to step S28 to store the designated associative word and sensory pattern weights designated in step S26 in the search weight holding unit **204**, thus ending the processing.

FIG. **20** is a flow chart showing the details of the associative word unfolding process **208** and the image content word search process **210** using associative words (step S4 in FIG. **18**).

Referring to FIG. **20**, the associative word dictionary **211** is searched using the query word stored in the query word holding unit **202** to obtain associative word data in step S41. More specifically, the associative word dictionary **211** is searched for index words **91** (FIG. **9**), which match the query word, and registered associative word data are extracted. If index words **91** that match the query word are found, all their associative IDs are stored in the unfolded associative word holding unit **209**.

In step S42, the concept discrimination dictionary **205** is searched, and if an index word that matches the query word in the query word holding unit **202** is found, a search perspective **81** (FIG. **8**) corresponding to that index word is extracted. The extracted search perspective is compared with that stored in the search perspective holding unit **203** in step S25, and if they match, an antithetic concept **82** corresponding to this index word is extracted. On the other hand, if the two search perspectives do not match, data in which the query word matches an index word continues to be searched for. Upon completion of checking for all the extracted search perspectives, the flow advances to step S43.

In step S43, the associative word dictionary **211** is searched for associative words having an index word, which matches the antithetic concept found in step S42. If an index word that match the antithetic concept is found, their associative IDs are stored in the unfolded associative word holding unit **209** by appending a status code indicating an antithetic concept thereto.

In step S44, associative words are extracted based on the associative IDs stored in the unfolded associative word holding unit **209**, and the image content word holding unit **219** is searched for image content words that match the associative words. The search results are stored in the search result holding unit **216**.

In this process, the associative IDs stored in the unfolded associative word holding unit **209** are extracted, and corresponding associative data are extracted with reference to the associative word dictionary **211**. Next, the association strengths **94** of the extracted associative data are extracted, and are set in a work memory ASCF (not shown) on the RAM **13**. In this case, if a status code indicating an antithetic concept is appended to a given associative ID extracted from the unfolded associative word holding unit **209**, the sign of the association strength is inverted to indicate a negative association strength. However, if the association strength is already a negative value, that associative data is discarded, and the next associative data is checked.

Then, an associative perspective corresponding to each associative ID is extracted, and is compared with that stored in the search perspective holding unit **203**. If the two perspectives match, a predetermined value α is set in a work memory VPF (not shown) assured on the RAM **13**. If they do not match, a value $\alpha \times 0.1$ is set in the work memory VPF.

Finally, the image content word holding unit **219** is searched for image content words that match associative words corresponding to the associative IDs. It an image content word is found, an image ID corresponding to that image content word is acquired from the image ID **72** (FIG. **7**), and is set in the found image ID **100** (FIG. **10**) in the search result holding unit **216**. "1" is set in the number **101** of matched associative words, and the found associative ID is set in the associative word ID **102**. Then, a value obtained

by multiplying the value in the work memories ASCF and VPF on the RAM **13** by a predetermined score β based on associative word matching is stored as an associative matching level in the associative matching level **103**. If an identical image ID has already been stored, the value of the numher **101** of matched associative words is incremented by 1, a new associative word ID is added to the associative word ID **102**, and the calculated associative matching level is added to the stored associative matching level **103** to update its value.

FIG. **21** is a flow chart showing the details of the sensory pattern unfolding process **212**, sensory pattern search process **214**, and search result integration process **217**.

As described above, the user inputs a search request for searching images by the search request input process **201**. The search request contains one or a plurality of query words, search perspectives, and the like. The query word input in this embodiment is an abstract image word that expresses impressions of images such as "refreshing", "warm", and the like. In this embodiment, assume that an image word "refreshing" is stored.

Steps **S61** and **S62** are implemented by the sensory pattern unfolding process **212**. In step **S61**, the image word held in the query word holding unit **202** is unfolded into sensory patterns with reference to the image word/sensory pattern correspondence holding unit **215**. In this embodiment, the query word holding unit **202** stores the image word "refreshing", the unfolded associative word holding unit **209** holds associative words "forest", "tableland", "blue sky", and the like unfolded from "refreshing", and the image word is unfolded into corresponding sensory pattern IDs with reference to the image word/sensory pattern correspondence holding unit **215**. For example, sensory pattem IDs "005" and "006" corresponding to image word "refreshing"—associative word "forest" are acquired, and a sensory pattern ID "007" corresponding to image word "refreshing"—associative word "tableland" is acquired.

The flow then advances to step **S62** to store the sets of unfolded sensory pattern IDs and image words/associative words in the unfolded sensory pattern holding unit **213**. The data storage in the unfolded sensory pattern holding unit **213** is as has already been described previously with reference to FIG. **11**.

The flow advances to step **S63**. Steps **S63** and **S64** are implemented by the sensory pattern search process **214**. In step **S63**, all image IDs of images having matching levels larger than zero with respect to the sensory pattern IDs stored in the unfolded sensory pattern holding unit **213** are acquired. This process is done for all the sensory patterns held in the unfolded sensory pattern holding unit **213**. Note that the sensory pattern search process **214** acquires image IDs having matching levels larger than zero with respect to the sensory pattern IDs respectively unfolded from the query word and antithetic concept.

In step **S64**, sets of acquired sensory pattern IDs, image IDs, and their matching levels acquired in step **S63** are stored in the search result holding unit **216**.

The flow then advances to step **S65**. Steps **S65** to **S69** are implemented by the search result integration process **217**. That is, two sets of search results, i.e., the image content word search results using associative words and sensory pattern search results, which are held in the search result holding unit **216**, are integrated into one set of search results on the hasis of the search weights stored in the determined weight holding unit **207** with reference to those search results. When the sensory pattern search results include a sensory pattern hased on the antithetic concept to the query word, the corresponding image is excluded from the integrated results. Or the sensory pattern matching level of an image including an sensory pattern of the antithetic concept may be lowered upon integration.

In this integration process, a method of ohtaining common elements of two sets of search results in units of associative words (ANDing search results), a method of calculating integrated matching levels based on the weights on the searches, and selecting appropriate search results in descending order of integrated matching levels, and the like are available. In this embodiment, the method of calculating the integrated matching levels will be exemplified helow.

Let A he the associative matching level of an image that matches, e.g., an associative word "forest" stored in the search result holding unit **216**, B be the sensory matching level of an image that matches the sensory pattern ID "005" corresponding to the associative word "forest", and w1 and w2 (w1+w2=1) be the search weights stored in the determined weight holding unit **207**. Then, the integrated matching level is given by:

$$\textit{Integrated matching level}=w1\times A+w2\times B$$

or

$$\textit{Integrated matching level}=(w1\times A^2+w2\times B^2)^{1/2}$$

The integrated matching levels of all sensory patterns of all associative words are calculated. When one image ID has matching levels larger than zero with respect to a plurality of sensory pattern IDs, a plurality of integrated matching levels are obtained for one image. However, in this case, an image with the highest integrated matching level is adopted as a search result (step **S65**).

This process is done for all images corresponding to either set of search results larger than zero, and images whose integrated matching levels are larger than a predetermined threshold value X are selected as integrated search results (steps **S66**, **S67**, and **S68**).

In step **S69**, the sets of image IDs and their integrated matching levels are stored in the search result holding unit **216**, thus ending the search process.

An image registration process for registering test images will be explained helow with reference to FIG. **22**.

FIG. **22** is a flow chart showing an example of a search pre-process executed upon registering images. This process is controlled in accordance with a processing program stored in the DISK **14**.

In step **S81**, the user designates an image to be registered. The image to be registered is designated from those stored in an external storage device, an image input device, an image database server connected to this image processing apparatus, or the like (none of them are shown). In this embodiment, assume that images serving as test images are stored in advance, and the image to be registered is selected from them.

The flow then advances to step **S82**. In step **S82**, an image ID corresponding to the designated image file name, and various kinds of image information required for registration are acquired, and are supplied to the image feature extraction process **221**. The image ID is stored in correspondence with the image file name to manage an image, and is acquired by searching data in the image holding unit **218** using the image file name. Various kinds of information of the image include pixel values indicating the width and height of an image, the number of hits per pixel, the image size (in units of bytes),

the address of the area where a bitmap image is actually stored, and the like, for example, when the file format of this image is the bitmap format. Since these pieces of image information are stored in the header field of the image file, they can be acquired by referring to the header field. Even when the file format of the image is not the bitmap format but JFIF or FlashPix, required information can be similarly obtained from the header field of a file. Or the image holding unit **218** may store such image information, and the image information may be acquired by referring to the image holding unit **218** upon registration.

The flow advances to step **S83**. Step **S83** is implemented by the image feature amount extraction process **221**, and extracts physical image feature amounts by analyzing the image information corresponding to the designated image ID. FIG. **15** above shows an example of the image feature amounts in this embodiment, and representative colors are extracted in units of image regions/blocks. The representative color may be obtained by using a scheme of analyzing an actual bitmap image using various kinds of input image information in units of pixels, and calculating the average value of color components (values in a color space such as RGB, HVC, or the like) used in each region or block, or a color component with the highest frequency of occurrence as a representative color.

The flow advances to step **S84**. In step **S84**, image feature amounts **c1** to cn extracted in step **S83** are stored in the image feature amount holding unit **222** in correspondence with the image ID of this image. The data storage format in this case is as has already been described previously with reference to FIG. **16**.

The flow advances to step **S85**, and all sensory pattern IDs stored in the image feature amount/sensory pattern correspondence holding unit **223**, and image feature amounts corresponding to those sensory patterns are acquired with reference to the image feature amount/sensory pattern correspondence holding unit **223**. In this embodiment, the chromatic feature amounts correspond to the individual sensory patterns, as has already been described previously with reference to FIG. **17**.

The flow advances to step **S86**, and the matching level between each sensory pattern acquired in step **S85** and the image feature amounts corresponding to this image is calculated. This process is done by the sensory pattern determination process **224**. That is, the chromatic image feature amounts corresponding to each of the sensory patterns acquired in step **S85** are compared with the image feature amounts extracted in step **S83** to calculate their matching level. The matching levels for all sensory patterns stored in the image feature amount/sensory pattern correspondence holding unit **223** are calculated. The matching level is calculated using a scheme such as vector computations, statistic processes, or the like using cosine measure.

The flow advances to step **S87**. In step **S87**, the matching levels between all the sensory patterns and the image calculated in step **S86** are stored in the sensory pattern holding unit **220** in correspondence with the image ID of this image. The image storage example in the sensory pattern holding unit **220** is as has already been described previously with reference to FIG. **13**.

The aforementioned process is done for all images to be registered.

As described above, according to this embodiment, a search using feature amount data of multimedia information itself and a search using a content word appended to multimedia information are made on the basis of associative words, which are associated with a query word, and final search results can be obtained from the results of the two search processes. For this reason, desired image information can be accurately extracted.

As described above, according to this embodiment, since a search request (query word, search perspective, and the like) which is to be considered upon searching for desired multimedia information can be designated, an appropriate search can be made in accordance with the designated search request, and desired image information can be accurately extracted.

According to this embodiment, upon obtaining search results by integrating search results obtained by a search using keywords appended to images, and those obtained by a search using feature amount data of images themselves, since the weight ratios on the two search processes can be changed in correspondence with a query word, the image information wanted can be accurately extracted. For example, when a keyword "happy" is input as a search request, it is hard to associate it with image feature amounts since its meaning is essentially lexical. Hence, if a search that attaches importance on the image feature amount is made, images which do not match the searcb request are highly likely to be presented. On the other hand, for example, when a keyword "showy" is input as a search request, the keyword "showy" is more likely to evoke meanings measurable as image feature amounts. For this reason, if a search is made while attaching importance on content words appended to images, images which do not match the search request indicated by the input keyword are highly likely to be presented. Or actually "showy" images may be excluded from the search results. By contrast, according to this embodiment, when a query word "happy" is set via the user interface shown in FIG. **4**, heavier weights are set on associative words; when a query word "showy" is set, heavier weights are set on sensory patterns, thus making an accurate search with respect to either query word. of course, when the associated weight **83** and sensory pattern weight **84** in the concept discrimination dictionary **205** are appropriately set, an appropriate search can be made by only instructing to "use default weight values" on the user interface shown in FIG. **4**.

In the above embodiment, image information is used as stored information serving as test images. As for multimedia information (e.g., audio information) other than image information, the present invention can be applied by executing information feature amount extraction, and pairing the extracted information feature amount with sensory patterns.

In the above description, the image holding unit **218**, image content word holding unit **219**, and sensory pattern holding unit **220** which undergo a search are allocated on the DISK **14** that builds a single device, but these building components may be distributed on different devices, and processes may be done on the network via the NIC **19**.

Note that the present invention may be applied to either a system constituted by a plurality of devices (e.g., a host computer, an interface device, a reader, a printer, and the like), or an apparatus consisting of a single equipment (e.g., a copying machine, a facsimile apparatus, or the like).

The objects of the present invention are also achieved by supplying a storage medium, which records a program code of a software program that can implement the functions of the above-mentioned embodiments to the system or apparatus, and reading out and executing the program code stored in the storage medium by a computer (or a CPU or MPU) of the system or apparatus.

In this case, the program code itself read out from the storage medium implements the functions of the above-

mentioned embodiments, and the storage medium which stores the program code constitutes the present invention.

As the storage medium for supplying the program code, for example, a floppy disk, hard disk, optical disk, magneto-optical disk, CD-ROM, CD-R, magnetic tape, nonvolatile memory card, ROM, and the like may be used.

The functions of the above-mentioned embodiments may be implemented not only by executing the readout program code by the computer but also by some or all of actual processing operations executed by an OS (operating system) running on the computer on the basis of an instruction of the program code.

Furthermore, the functions of the above-mentioned embodiments may be implemented by some or all of actual processing operations executed by a CPU or the like arranged in a function extension board or a function extension unit, which is inserted in or connected to the computer, after the program code read out from the storage medium is written in a memory of the extension board or unit.

As many apparently widely different embodiments of the present invention can be made without departing from the spirit and scope thereof, it is to be understood that the invention is not limited to the specific embodiments thereof except as defined in the appended claims.

What is claimed is:

1. An information search apparatus for searching information based on an input query word, comprising:

first search means for determining a query keyword on the basis of the query word, and searching information on the basis of the query keyword;

second search means for determining a feature amount corresponding to the query word, and searching information on the basis of the feature amount;

setting means for setting a search weight to be assigned to search results of said first and second search means; and

integration means for integrating search results obtained by said first and second search means in accordance with the search weight set by said setting means.

2. The apparatus according to claim 1, wherein the search weight includes a first weight corresponding to the search result of said first search means, and a second weight corresponding to the search result of said second search means, and

said integration means applies the first weight to a search matching level of each information as the search result of said first search means and the second weight to a search matching level of each information as the search result of said second search means to obtain an integrated search matching level, and obtains integrated search results on the basis of the integrated search matching level.

3. The apparatus according to claim 2, wherein said integration means selects a predetermined number of pieces of information in descending order of integrated search matching level, and determines the selected information as the integrated search results.

4. The apparatus according to claim 1, wherein said setting means allows a user to set desired weight ratios with respect to the search results of said first and second search means.

5. The apparatus according to claim 1, further comprising:

a weight dictionary which registers weights corresponding to said first and second search means in relation to the query word, and

wherein said setting means sets the weights with reference to said weight dictionary.

6. The apparatus according to claim 5, wherein said first search means derives an associative word associated with the query word, and uses the query word and the derived associative word as query keywords,

said weight dictionary registers weights in units of associative perspectives that connect query words and associative words in units of query words, and

said setting means sets the weights with reference to said weight dictionary on the basis of the query word and an associative perspective designated by a user.

7. The apparatus according to claim 1, wherein the information searched is image data,

said apparatus further comprises:

an image content word holding unit for storing the image data and content words which verbalize concepts expressed in the image data in correspondence with each other; and

an associative word dictionary for storing associative words associated with the content words, and

said first search means acquires an associative word corresponding to the query word from said associative word dictionary, and searches said image content word holding unit on the basis of the acquired associative word.

8. The apparatus according to claim 7, further comprising:

a concept discrimination dictionary for storing index words and antithetic concepts corresponding to the index words in correspondence with each other; and

input means for inputting the query word and a search perspective, and

wherein said first search means acquires an index word and antithetic concept corresponding to the query word from said concept discrimination dictionary on the basis of the query word and search perspective input by said input means, and acquires an associative word corresponding to the query word from said associative word dictionary on the basis of the acquired index word and antithetic concept.

9. The apparatus according to claim 1, further comprising:

a holding unit for storing associative words and sensory patterns in correspondence with each other, and

wherein said second search means acquires a sensory pattern corresponding to an associative word, which corresponds to the query word, from said holding unit, and extracts a feature amount of the acquired sensory pattern as the feature amount corresponding to the query word.

10. The apparatus according to claim 1, wherein multimedia information is image information, and the feature amount is a physical image feature amount obtained by analyzing the image information.

11. The apparatus according to claim 10, wherein the feature amount includes at least one of color scheme information, composition information, and shape information contained of an image.

12. An information search method for searching information based on an input query word, comprising:

a first search step, of determining a query keyword on the basis of the query word, and searching information on the basis of the query keyword;

a second search step, of determining a feature amount corresponding to the query word, and searching information on the basis of the feature amount;

a setting step, of setting a search weight to be assigned to search results in the first and a second search steps; and

an integration step, of integrating search results obtained in the first and a second search steps in accordance with the search weight set in the setting step.

13. The method according to claim 12, wherein the search weight includes a first weight corresponding to the search result in the first search step, and a second weight corresponding to the search result in the second search step, and

the integration step includes the step of applying the first weight to a search matching level of each information as the search result in the first search step and the second weight to a search matching level of each information as the search result in the second search step to obtain an integrated search matching level, and obtaining integrated search results on the basis of the integrated search matching level.

14. The method according to claim 13, wherein the integration step includes the step of selecting a predetermined number of pieces of information in descending order of integrated search matching level, and determining the selected information as the integrated search results.

15. The method according to claim 12, wherein the setting step includes the step of allowing a user to set desired weight ratios with respect to the search results in the first and second search steps.

16. The method according to claim 12, wherein the setting step includes a step of setting the weights with reference to a weight dictionary which registers weights corresponding to the first and second search steps in relation to the query word.

17. The method according to claim 16, wherein the first search step includes a step of deriving an associative word associated with the query word, and using the query word and the derived associative word as query keywords,

the weight dictionary registers weights in units of associative perspectives that connect query words and associative words in units of query words, and

the setting step includes a step of setting the weights with reference to the weight dictionary on the basis of the query word and an associative perspective designated by a user.

18. The method according to claim 12, wherein the information searched is image data, and said method is performed using:

an image content word holding unit for storing the image data and content words which verbalize concepts expressed in the image data in correspondence with each other; and

an associative word dictionary for storing associative words associated with the content words, and

wherein the first search step includes a step of acquiring an associative word corresponding to the query word from the associative word dictionary, and searching the image content word holding unit on the basis of the acquired associative word.

19. The method according to claim 18, wherein said method is performed using a concept discrimination dictionary for storing index words and antithetic concepts corresponding to the index words in correspondence with each other; and

wherein said method further comprises an input step, of inputting the query word and a search perspective, and

wherein the first search step includes a step of acquiring an index word and antithetic concept corresponding to the query word from the concept discrimination dictionary on the basis of the query word and search perspective input in the input step, and acquiring an associative word corresponding to the query word from the associative word dictionary on the basis of the acquired index word and antithetic concept.

20. The method according to claim 12, wherein said method is performed using a holding unit for storing associative words and sensory patterns in correspondence with each other, and

wherein the second search step includes a step of acquiring a sensory pattern corresponding to an associative word, which corresponds to the query word, from the holding unit, and extracting a feature amount of the acquired sensory pattern as the feature amount corresponding to the query word.

21. The method according to claim 12, wherein multimedia information is image information, and the feature amount is a physical image feature amount obtained by analyzing the image information.

22. The method according to claim 21, wherein the feature amount includes at least one of color scheme information, composition information, and shape information contained of an image.

23. A storage medium for storing a control program which makes a computer search information based on an input query word, said control program comprising:

a code of the first search step of determining a query keyword on the basis of the query word, and searching information on the basis of the query keyword;

a code of the second search step of determining a feature amount corresponding to the query word, and searching information on the basis of the feature amount;

a code of the setting step of setting a search weight to be assigned to search results in the first and second search steps; and

a code of the integration step of integrating search results obtained in the first and second search steps in accordance with the search weight set in the setting step.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 6,526,400 B1
DATED : February 25, 2003
INVENTOR(S) : Takata et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Column 5,
Line 18, "an" should read -- and --.

Column 6,
Line 33, ".holding" should read -- holding --.

Column 8,
Line 20, "no," should read -- no --.

Column 18,
Line 36, "of" should read -- Of --.

Signed and Sealed this

Ninth Day of December, 2003

JAMES E. ROGAN
Director of the United States Patent and Trademark Office

US006493705B1

(12) **United States Patent**
Kobayashi et al.

(10) **Patent No.: US 6,493,705 B1**
(45) **Date of Patent: Dec. 10, 2002**

(54) **INFORMATION SEARCH APPARATUS AND METHOD, AND COMPUTER READABLE MEMORY**

(75) Inventors: **Yuji Kobayashi**, Kawasaki (JP); **Tomomi Takata**, Yokohama (JP)

(73) Assignee: **Canon Kabushiki Kaisha**, Tokyo (JP)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/409,056**

(22) Filed: **Sep. 30, 1999**

(30) **Foreign Application Priority Data**

Sep. 30, 1998 (JP) 10-278722
Sep. 30, 1998 (JP) 10-278724
Sep. 30, 1998 (JP) 10-278728

(51) **Int. Cl.**[7] **G06F 17/30**
(52) **U.S. Cl.** **707/3**; 707/10; 707/104.1; 707/513
(58) **Field of Search** 707/1–5, 104.1, 707/500, 8, 10, 513

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,778,368	A		7/1998	Hogan et al. 707/10
5,842,206	A	*	11/1998	Sotomayor 707/5
5,884,309	A	*	3/1999	Vanechanos, Jr. 706/50
5,893,092	A	*	4/1999	Driscoll 707/5
5,893,095	A		4/1999	Jain et al. 707/6
5,983,237	A	*	11/1999	Jain et al. 707/104.1
6,078,916	A	*	6/2000	Culliss 707/5
6,085,186	A		7/2000	Christianson et al. 707/3
6,088,692	A	*	7/2000	Driscoll 707/5
6,102,969	A		8/2000	Christianson et al. 717/8
6,122,628	A		9/2000	Castelli et al. 707/5
6,122,647	A		9/2000	Horowitz et al. 707/513
6,175,829	B1	*	1/2001	Li et al. 707/3
6,182,068	B1	*	1/2001	Culliss 707/5
6,226,636	B1		5/2001	Abdel-Mottaleb et al. 707/4
6,249,784	B1	*	6/2001	Macke et al. 707/3
6,347,313	B1	*	.2/2002	Ma et al. 707/3

OTHER PUBLICATIONS

Mukherjea, Sougata et al., "Visualizing World-Wide Web Search Engine Results", Proceedings of the 1999 IEEE International Conference on Information Visualization, Jul. 14–16, 1999, pp. 400–405.*

Yanai, Keiji, "An-Automatic Image-Gathering System for the World-Wide Web by Integration of Keywords and Image Features", Proceedings, Forth International Conference on Computational Intelligence and Multimedia Applications, Oct. 2001, pp. 303–307.*

Chung-Sheng Li, Mohan, R., Smith, J.R., Multimedia Content desription in the InfoPyramid,. Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing vol. 6, May 12–15, 1998, pp. 3789–3792.

Hjelsvold, R., Midtstraum, R., Sandsta, O., Searching and browsing a Shared Video Database, Proceedings., International Workshop on Multi-Media Database Management Systems, 1995., Aug. 28–30, 1995, pp. 90–98.

Wittenburg, T.M., Little, T.D.C., An adaptive document management System for Shared Multimedia Data. Proceedings of the International Conference on Multimedia Computing and Systems, 1994., May 15–19, 1994; pp. 245–254.

* cited by examiner

Primary Examiner—Jean M. Corrielus
Assistant Examiner—Shahid Alam
(74) *Attorney, Agent, or Firm*—Fitzpatrick, Cella, Harper & Scinto

(57) **ABSTRACT**

A search request input processing unit holds an input query word in a query word holding unit. An associative word unfolding processing unit acquires associative words in association with the query word held in the query word holding unit with reference to an associative word dictionary. An image content search processing unit using associative words makes a keyword search of image information on the basis of the obtained associative words and the query word. A sensory pattern unfolding processing unit obtains the query word and sensory patterns corresponding to associative words of the query word, and searches image information using feature amounts of the obtained sensory patterns. A search result integration processing unit integrates the search results obtained by the image content word search processing unit and sensory pattern search processing unit.

26 Claims, 22 Drawing Sheets

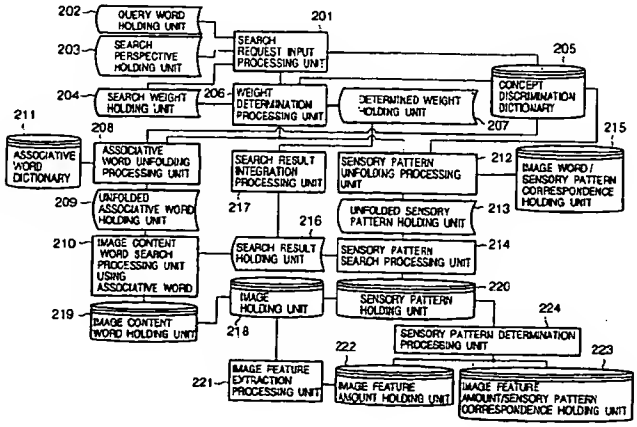


FIG. 1

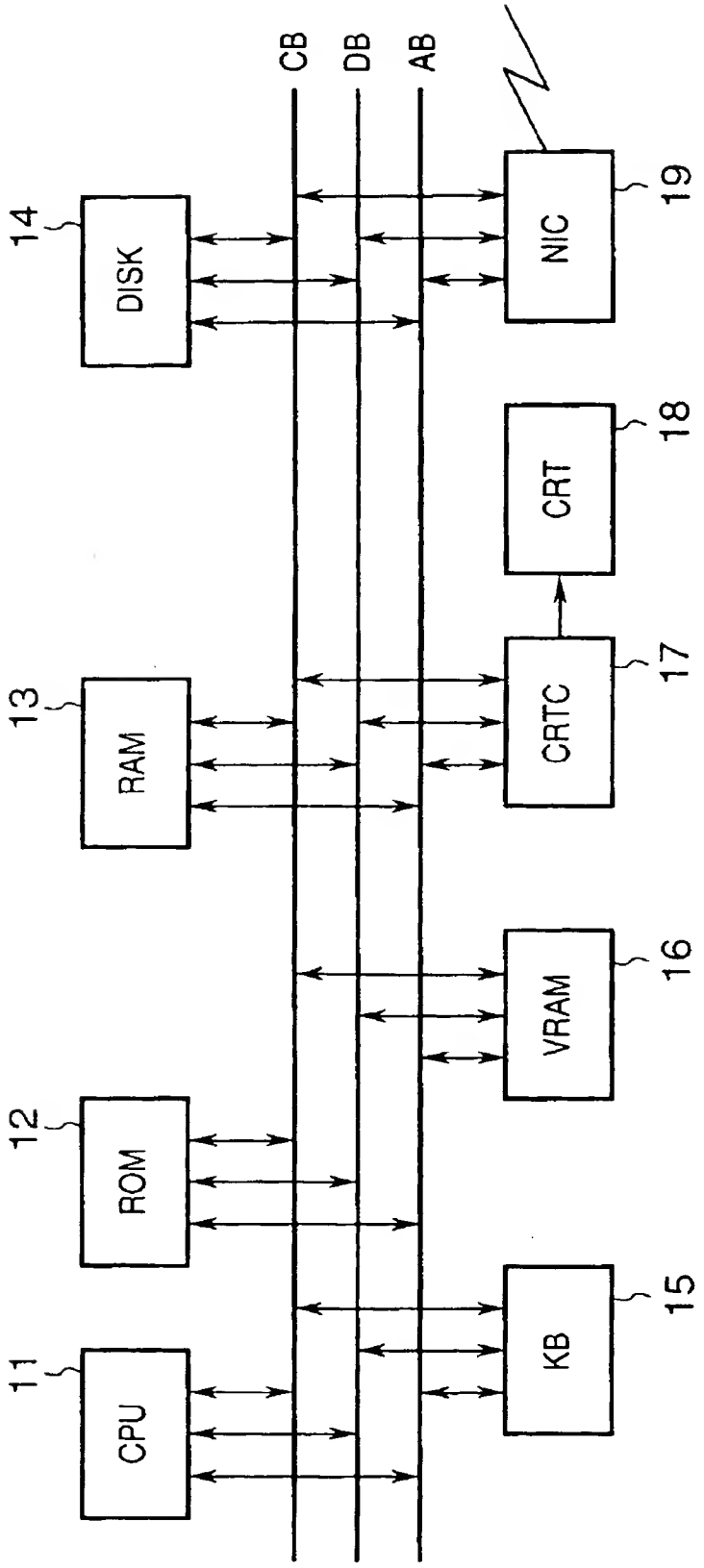

CB
DB
AB
DISK
14
NIC
19
CRT
18
RAM
13
CRTC
17
VRAM
16
ROM
12
KB
15
CPU
11

FIG. 2

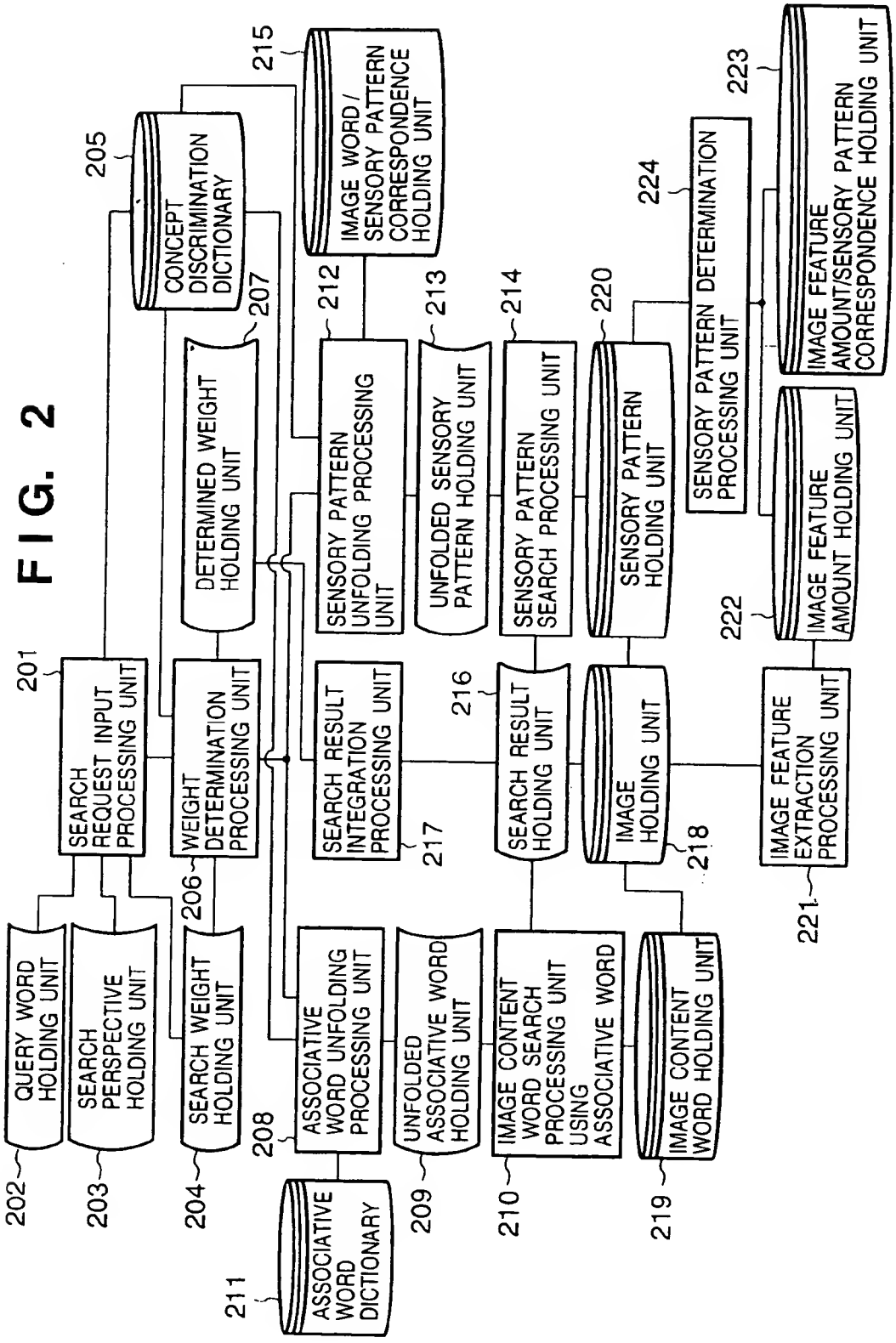

201
SEARCH REQUEST INPUT PROCESSING UNIT
202
QUERY WORD HOLDING UNIT
203
SEARCH PERSPECTIVE HOLDING UNIT
204
SEARCH WEIGHT HOLDING UNIT
205
CONCEPT DISCRIMINATION DICTIONARY
206
WEIGHT DETERMINATION PROCESSING UNIT
207
DETERMINED WEIGHT HOLDING UNIT
208
ASSOCIATIVE WORD UNFOLDING PROCESSING UNIT
209
UNFOLDED ASSOCIATIVE WORD HOLDING UNIT
210
IMAGE CONTENT WORD SEARCH PROCESSING UNIT USING ASSOCIATIVE WORD
211
ASSOCIATIVE WORD DICTIONARY
212
SENSORY PATTERN UNFOLDING PROCESSING UNIT
213
UNFOLDED SENSORY PATTERN HOLDING UNIT
214
SENSORY PATTERN SEARCH PROCESSING UNIT
215
IMAGE WORD/ SENSORY PATTERN CORRESPONDENCE HOLDING UNIT
216
SEARCH RESULT HOLDING UNIT
217
SEARCH RESULT INTEGRATION PROCESSING UNIT
218
IMAGE HOLDING UNIT
219
IMAGE CONTENT WORD HOLDING UNIT
220
SENSORY PATTERN HOLDING UNIT
221
IMAGE FEATURE EXTRACTION PROCESSING UNIT
222
IMAGE FEATURE AMOUNT HOLDING UNIT
223
IMAGE FEATURE AMOUNT/SENSORY PATTERN CORRESPONDENCE HOLDING UNIT
224
SENSORY PATTERN DETERMINATION PROCESSING UNIT

FIG. 3

IMAGE WORD (2150-1)	ASSOCIATIVE WORD (2150-2)	SENSORY PATTERN ID (2150-3)
REFRESHING	FOREST	005
REFRESHING	FOREST	006
REFRESHING	TABLELAND	007
REFRESHING	BLUE SKY	012
REFRESHING	BLUE SKY	015
REFRESHING	SEA	016
REFRESHING		020
NOT REFRESHING		001
NOT REFRESHING		010
TROPICAL	SEA	023
TROPICAL	FRUIT	052
TROPICAL	FRUIT	053
TROPICAL	...	...
...	...	...

FIG. 4

IMAGE WORD (2130-1)	R E F R E S H I N G NULL ...
NUMBER OF SENSORY PATTERNS (2130-2)	7
STORAGE LOCATION ADDRESS OF PATTERN INFORMATION (2130-3)	

2130-4

SENSORY PATTERN ID (2130-5)	5
ASSOCIATIVE WORD (2130-6)	F O R E S T NULL ...
SENSORY PATTERN ID	12
ASSOCIATIVE WORD	B L U E S K Y NULL ...
...	...

FIG. 5

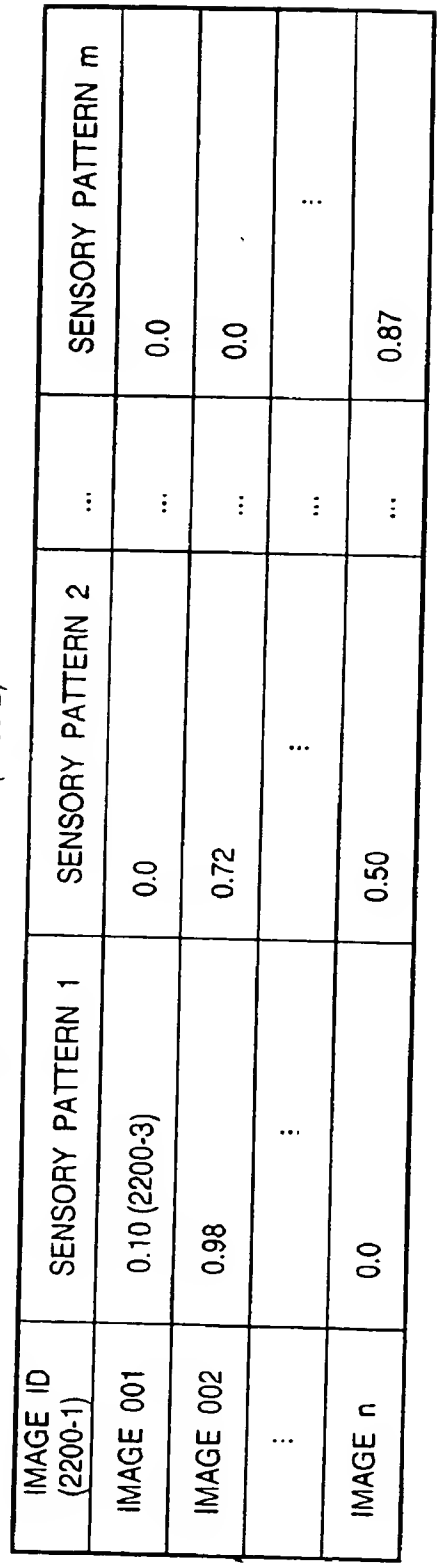

SENSORY PATTERN ID (2200-2)
IMAGE ID (2200-1)
SENSORY PATTERN 1
SENSORY PATTERN 2
...
SENSORY PATTERN m
IMAGE 001
0.10 (2200-3)
0.0
0.0
IMAGE 002
0.98
0.72
0.0
IMAGE n
0.0
0.50
0.87

FIG. 6

PHYSICAL IMAGE FEATURE (X_1~X_n)

	X_1	X_2	X_3	X_4	X_5	X_6	···	Xn
B1	x11	x12	X13	X14	X15	X16		X1n
B2	x21	x22	x23	x24	x25	x26		x2n
B3	x31	x32	x33	x34	x35	x36		x3n
B4	x41	x42	x43	x44	x45	x46		x4n
B5	x51	x52	x53	x54	x55	x56		x5n
⋮								
Bm	xm1	xm2	xm3	xm4	xm5	xm6		xmn

B1···Bm: REGION OR BLOCK NO.

FIG. 7

CHROMATIC FEATURE AMOUNT (2230-2)

SENSORY PATTERN ID (2230-1)	COLOR 1			COLOR 2			...	COLOR m		
	R	G	B	R	G	B		R	G	B
PATTERN 1	255	0	0	255	255	0		0	0	0
PATTERN 2	153	153	61	255	255	255		161	179	89
...										
PATTERN n	102	255	255	255	255	255		102	179	255

FIG. 8

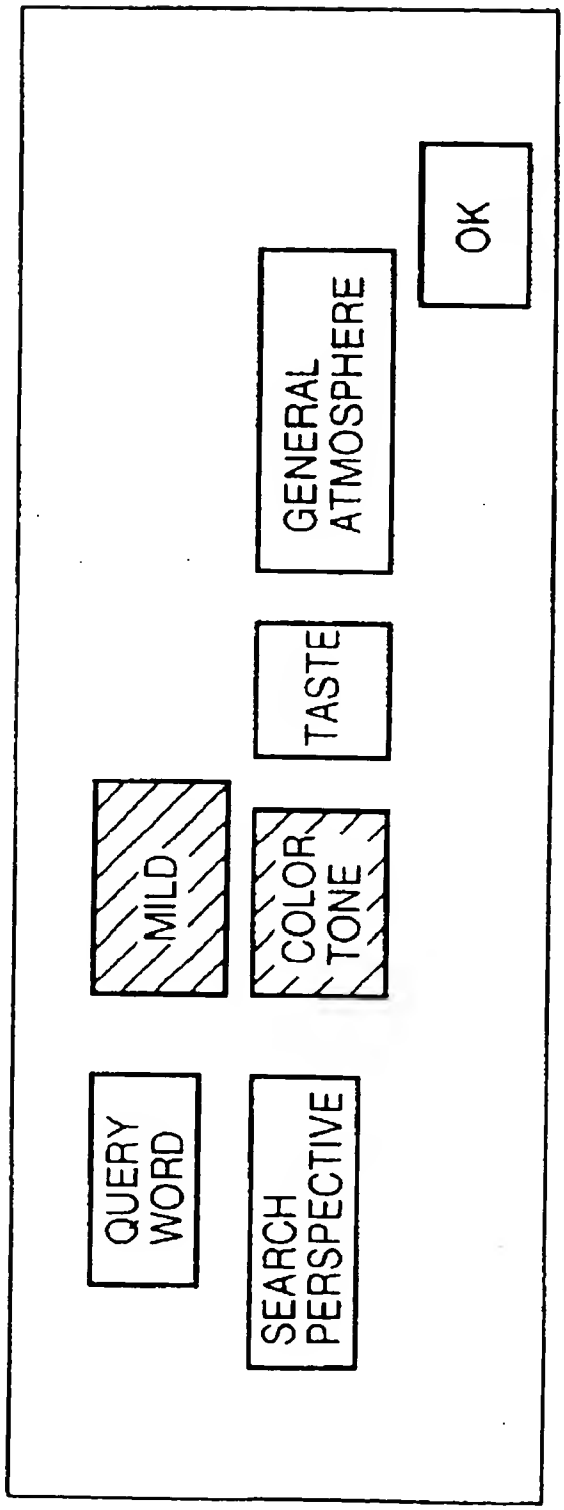

OK
GENERAL ATMOSPHERE
TASTE
COLOR TONE
MILD
SEARCH PERSPECTIVE
QUERY WORD

FIG. 9

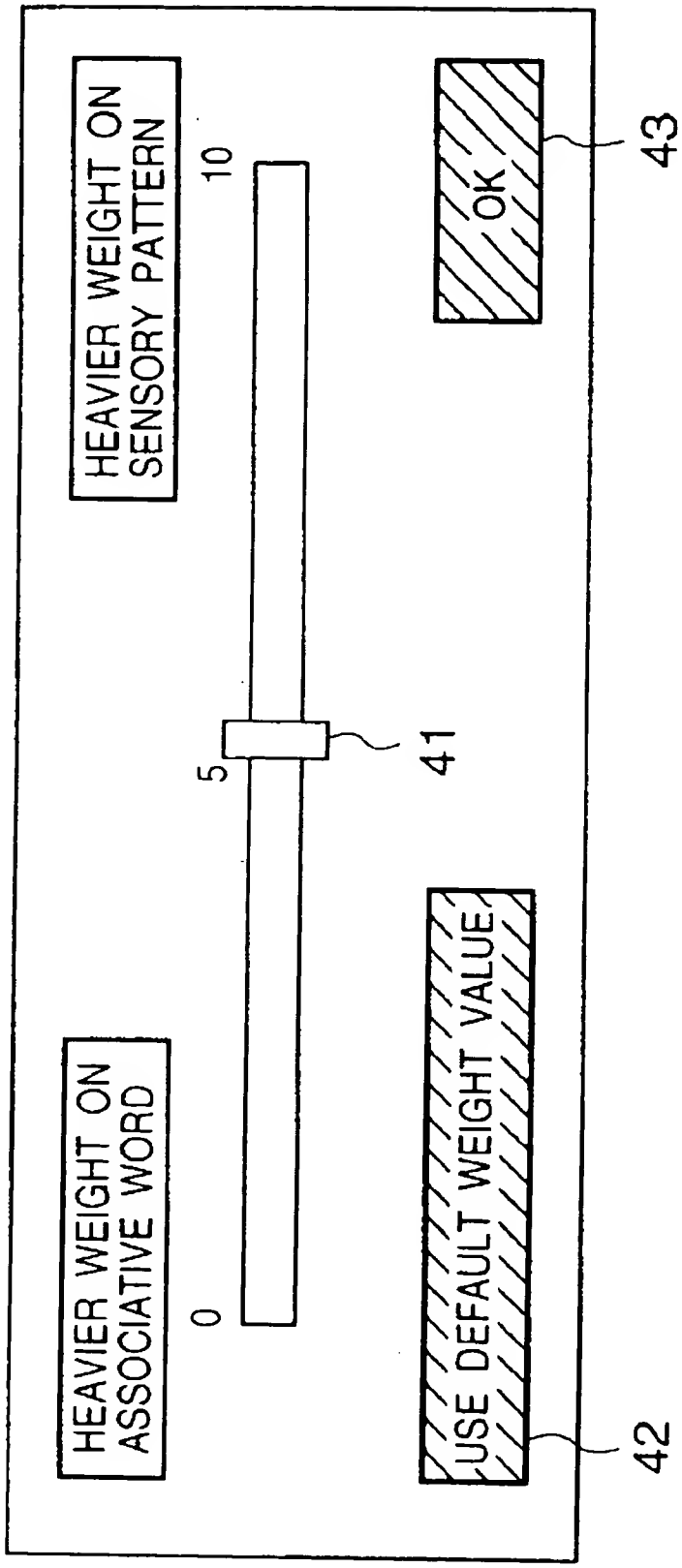

HEAVIER WEIGHT ON SENSORY PATTERN
10
5
0
41
OK
43
HEAVIER WEIGHT ON ASSOCIATIVE WORD
USE DEFAULT WEIGHT VALUE
42

FIG. 10

2180 2181

IMAGE ID	IMAGE FILE PATH
...	...
1024	X : ¥SCENE¥123. BMP
...	...
1563	X : ¥HUMAN¥078. BMP
1564	X : ¥HUMAN¥079. BMP
1565	X : ¥HUMAN¥080. BMP
...	...

FIG. 11

21900 21901

IMAGE ID	IMAGE CONTENT WORD
...	...
1024	TABLELAND, BLUE SKY, CLOUD, MOUNTAIN
1025	SKY, SUN, CLOUD
1026	SKY, SEA, HORIZON
...	...

FIG. 12

21910 21911

IMAGE CONTENT WORD	IMAGE ID
...	...
BLUE SKY	988, 1020, 1024
SEA	867, 888, 1026, 1347
CLOUD	991, 1024, 1025, 1078
TABLELAND	1024
SKY	777, 778, 1025, 1026
SUN	1025, 1066
MOUNTAIN	1024, 1111, 1112, 1113, 1115
...	...

FIG. 13

2050 INDEX WORD	2051 SEARCH PERSPECTIVE	2052 ANTITHETIC CONCEPT	2053 ASSOCIATED WEIGHT	2054 SENSORY PATTERN WEIGHT
...	...	...	...	...
REFRESHING	CHARACTER DESCRIPTION	SHABBY	8	2
REFRESHING	LANDSCAPE	HEAVY	3	7
REFRESHING	TASTE	THICK	9	1
REFRESHING	GENERAL ATMOSPHERE		5	5
...	...	...	...	...
MILD	COLOR TONE	SHOWY	2	8
MILD	TASTE		9	1
MILD	GENERAL ATMOSPHERE		5	5
...	...	...	...	...
SIMPLE	GENERAL ATMOSPHERE	URBAN	5	5
SIMPLE	COLOR TONE	SHOWY	3	7
...	...	...	...	...
SMOOTH	FEEL	SANDY	7	3
SMOOTH	TEXTURE	SANDY	5	5
SMOOTH	GENERAL ATMOSPHERE	SANDY	6	4
...	...	...	...	...

FIG. 14

2110 ASSOCIATIVE ID	2111 INDEX WORD	2112 ASSOCIATIVE WORD	2113 ASSOCIATIVE PERSPECTIVE	2114 ASSOCIATION STRENGTH
...	...	...	...	...
58790	REFRESHING	ATHLETE	CHARACTER DESCRIPTION	6
58791	REFRESHING	NICE GUY	CHARACTER DESCRIPTION	5
58792	REFRESHING	SUMMER RESORT	LANDSCAPE	6
58793	REFRESHING	TABLELAND	LANDSCAPE	7
58794	REFRESHING	BLUE SKY	LANDSCAPE	8
58795	REFRESHING	LEMON	TASTE	6
58796	REFRESHING	SODA	TASTE	4
58797	REFRESHING	SHOWER	GENERAL ATMOSPHERE	5
...	...	...	...	...
71238	MILD	TEA	TASTE	7
71239	MILD	WINE	TASTE	2
71240	MILD	EARTH COLOR	COLOR TONE	4
...	...	...	...	...
126531	SIMPLE	COUNTRY	GENERAL ATMOSPHERE	5
126532	SIMPLE	LOCAL STYLE COOKING	GENERAL ATMOSPHERE	5
126533	SIMPLE	FOLKCRAFT ARTICLE	GENERAL ATMOSPHERE	6
126534	SIMPLE	SPACE SHUTTLE	GENERAL ATMOSPHERE	-8
126535	SIMPLE	SPORT CAR	GENERAL ATMOSPHERE	-8
126536	SIMPLE	CHANDELIER	GENERAL ATMOSPHERE	-9
...	...	...	...	...
234099	SMOOTH	VELVETY SKIN	FEEL	3
234100	SMOOTH	VELVET	FEEL	6
234101	SMOOTH	LUSTER	TEXTURE	5
...	...	...	...	...

FIG. 15

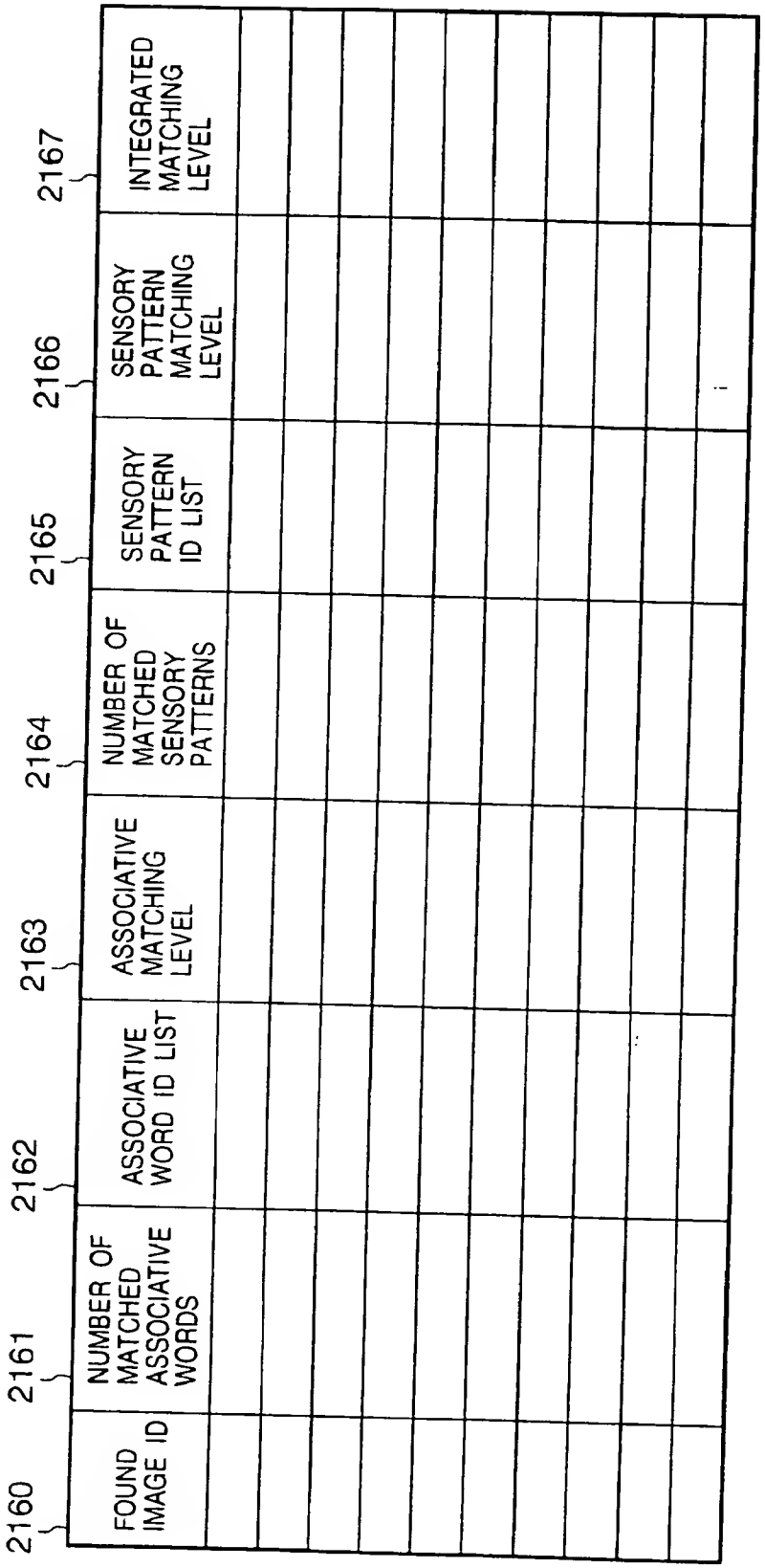

2160
FOUND IMAGE ID
2161
NUMBER OF MATCHED ASSOCIATIVE WORDS
2162
ASSOCIATIVE WORD ID LIST
2163
ASSOCIATIVE MATCHING LEVEL
2164
NUMBER OF MATCHED SENSORY PATTERNS
2165
SENSORY PATTERN ID LIST
2166
SENSORY PATTERN MATCHING LEVEL
2167
INTEGRATED MATCHING LEVEL

FIG. 16

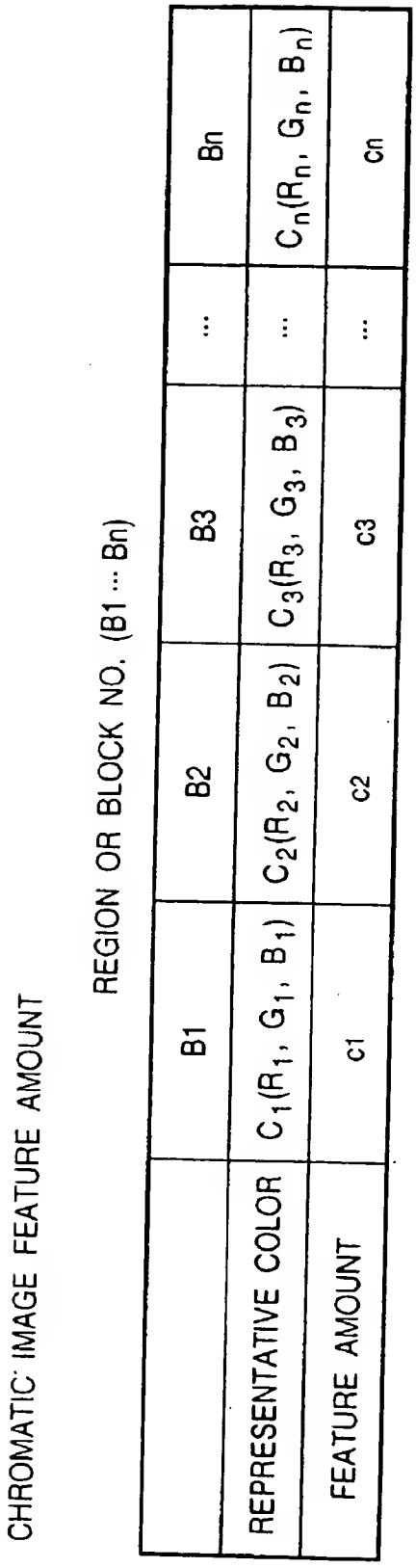

CHROMATIC IMAGE FEATURE AMOUNT
REGION OR BLOCK NO. (B1 ··· Bn)
B1
B2
B3
···
Bn
REPRESENTATIVE COLOR
$C_1(R_1, G_1, B_1)$
$C_2(R_2, G_2, B_2)$
$C_3(R_3, G_3, B_3)$
···
$C_n(R_n, G_n, B_n)$
FEATURE AMOUNT
c1
c2
c3
···
cn

FIG. 17

BLOCK OR REGION NO. (2220-2)

IMAGE ID (2220-1)	B1		...	Bm	
	REPRESENTATIVE COLOR (2220-3)	FEATURE AMOUNT (2220-4)	...	REPRESENTATIVE COLOR	FEATURE AMOUNT
IMAGE 001	C11 (R11, G11, B11)	c11	...	C1m (R1m, G1m, B1m)	c1m
IMAGE 002	C21 (R21, G21, B21)	c21	...	C2m (R2m, G2m, B2m)	c2m
...	...	...	...	...	...
IMAGE n	Cn1 (Rn1, Gn1, Bn1)	cn1	...	Cnm (Rnm, Gnm, Bnm)	cnm

FIG. 18

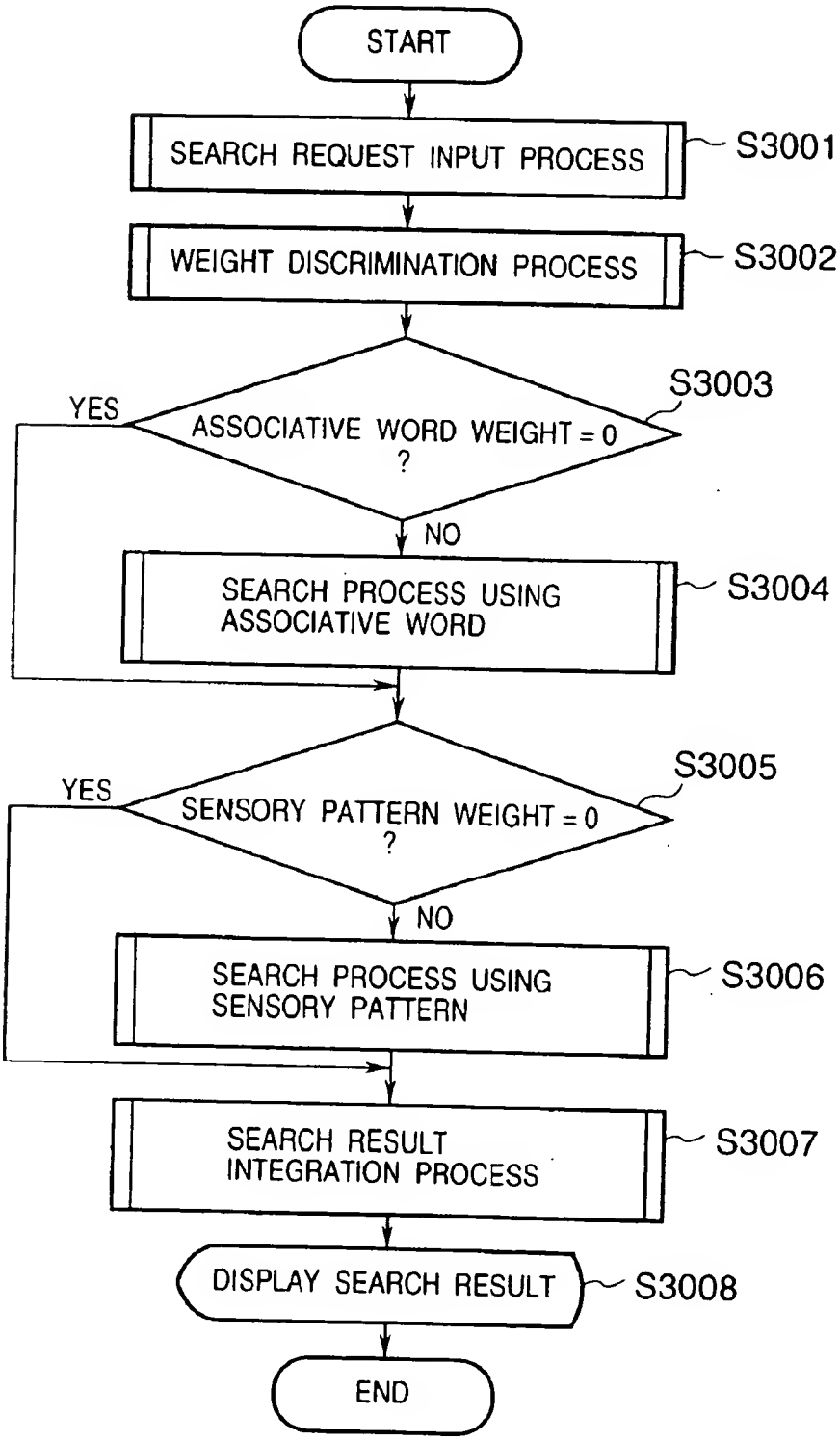

START
SEARCH REQUEST INPUT PROCESS
S3001
WEIGHT DISCRIMINATION PROCESS
S3002
ASSOCIATIVE WORD WEIGHT = 0 ?
S3003
YES
NO
SEARCH PROCESS USING ASSOCIATIVE WORD
S3004
SENSORY PATTERN WEIGHT = 0 ?
S3005
YES
NO
SEARCH PROCESS USING SENSORY PATTERN
S3006
SEARCH RESULT INTEGRATION PROCESS
S3007
DISPLAY SEARCH RESULT
S3008
END

FIG. 19

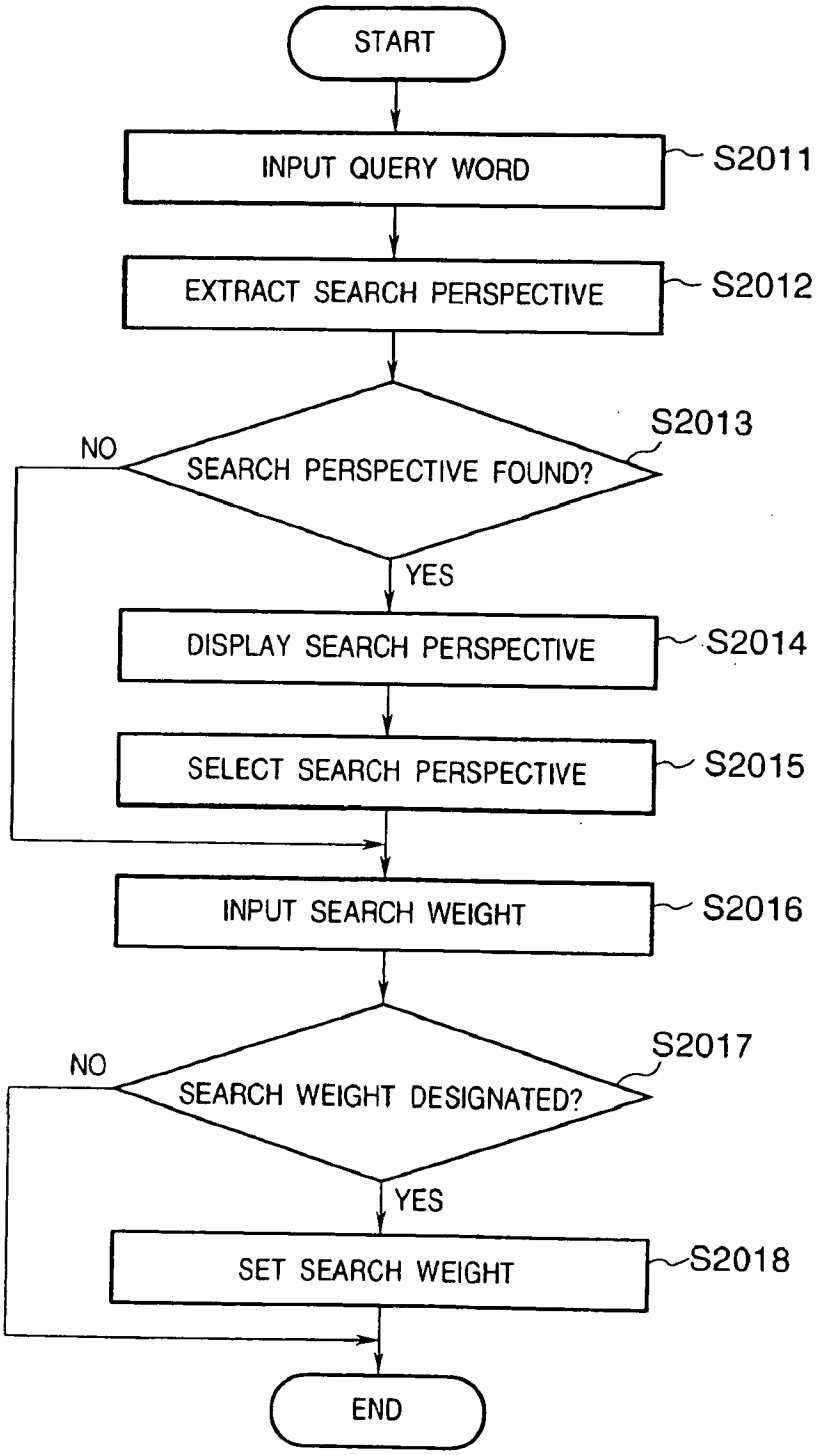

START
INPUT QUERY WORD
S2011
EXTRACT SEARCH PERSPECTIVE
S2012
SEARCH PERSPECTIVE FOUND?
S2013
NO
YES
DISPLAY SEARCH PERSPECTIVE
S2014
SELECT SEARCH PERSPECTIVE
S2015
INPUT SEARCH WEIGHT
S2016
SEARCH WEIGHT DESIGNATED?
S2017
NO
YES
SET SEARCH WEIGHT
S2018
END

FIG. 20

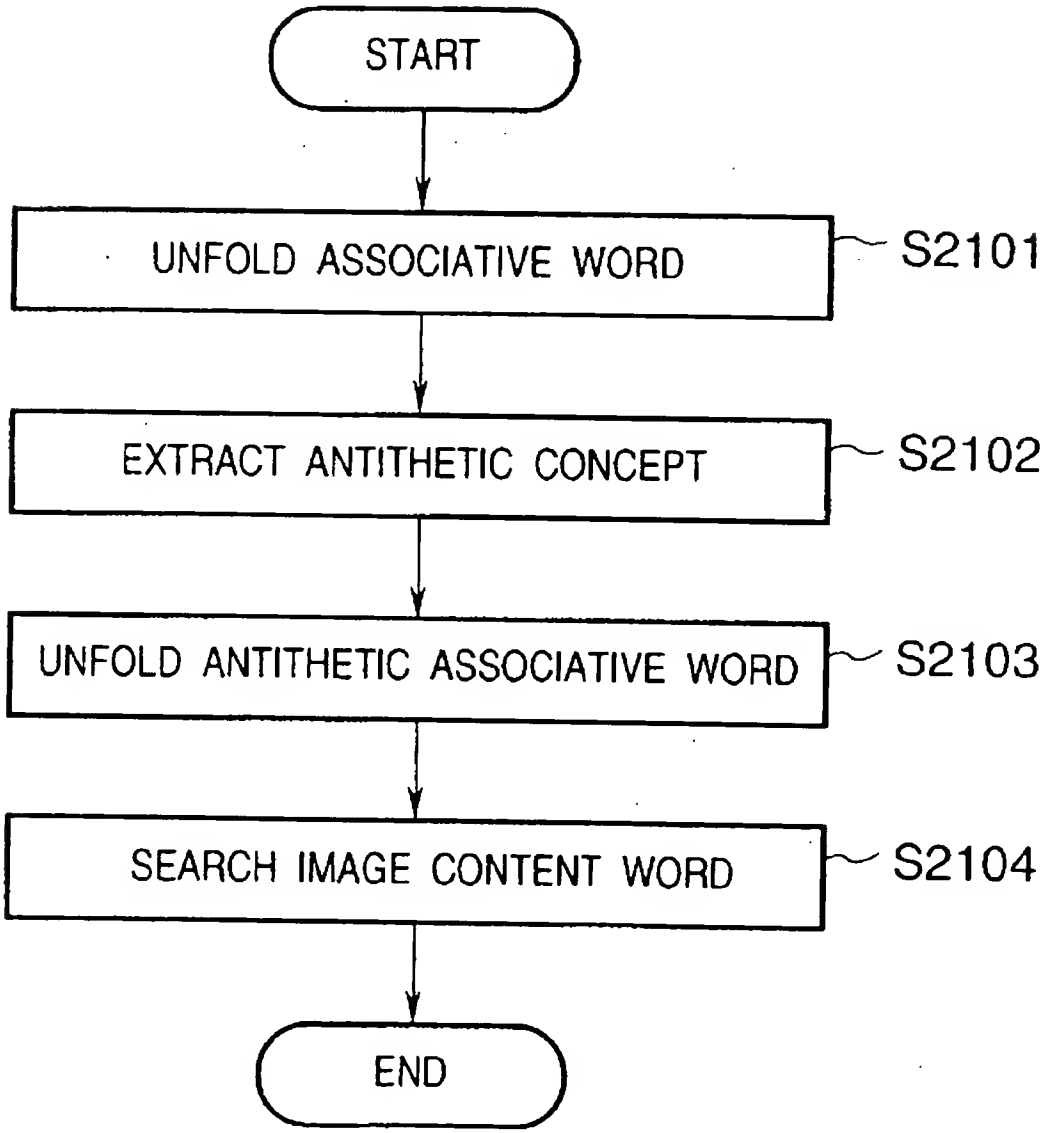

START
UNFOLD ASSOCIATIVE WORD
S2101
EXTRACT ANTITHETIC CONCEPT
S2102
UNFOLD ANTITHETIC ASSOCIATIVE WORD
S2103
SEARCH IMAGE CONTENT WORD
S2104
END

FIG. 21

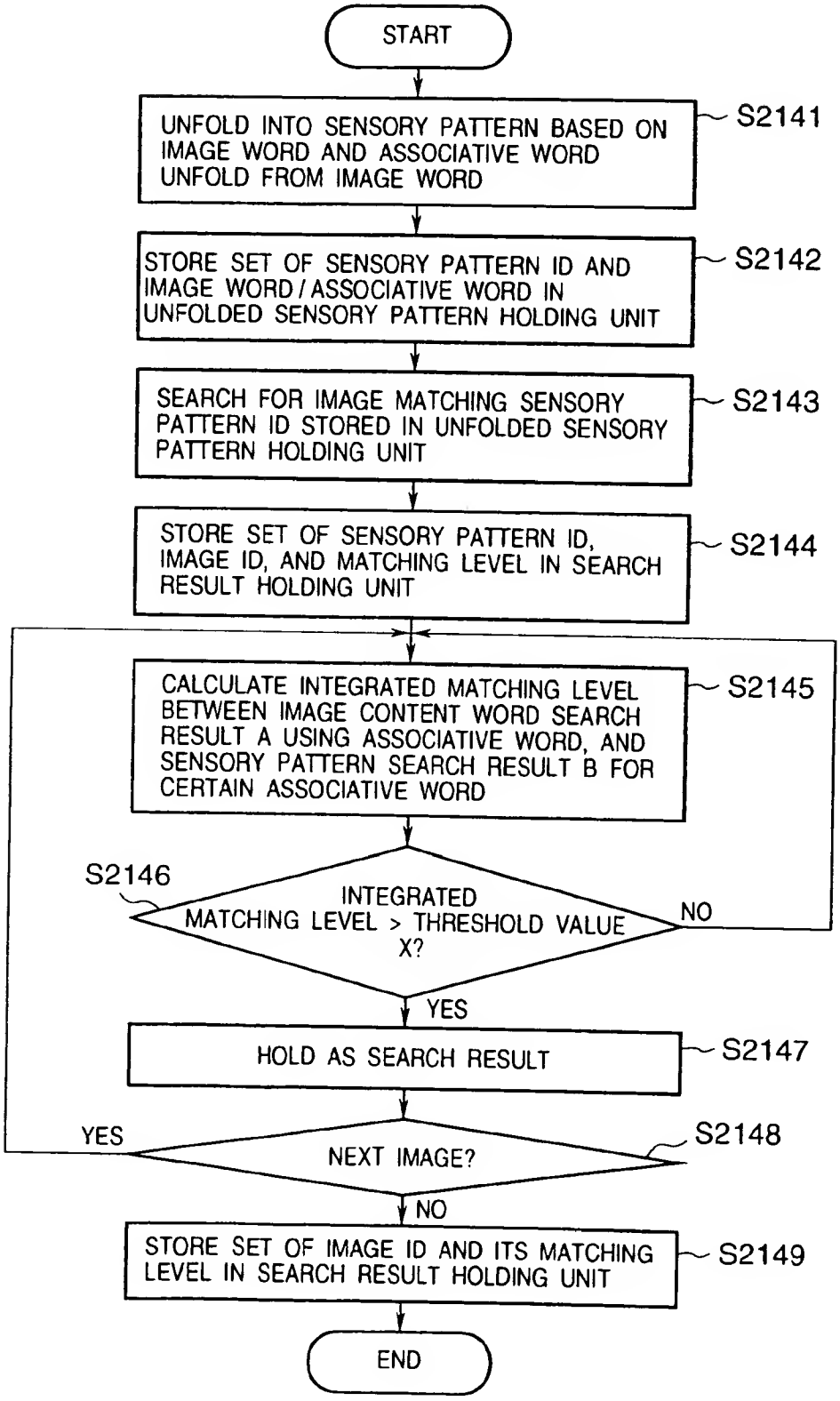

START
UNFOLD INTO SENSORY PATTERN BASED ON IMAGE WORD AND ASSOCIATIVE WORD UNFOLD FROM IMAGE WORD
S2141
STORE SET OF SENSORY PATTERN ID AND IMAGE WORD / ASSOCIATIVE WORD IN UNFOLDED SENSORY PATTERN HOLDING UNIT
S2142
SEARCH FOR IMAGE MATCHING SENSORY PATTERN ID STORED IN UNFOLDED SENSORY PATTERN HOLDING UNIT
S2143
STORE SET OF SENSORY PATTERN ID, IMAGE ID, AND MATCHING LEVEL IN SEARCH RESULT HOLDING UNIT
S2144
CALCULATE INTEGRATED MATCHING LEVEL BETWEEN IMAGE CONTENT WORD SEARCH RESULT A USING ASSOCIATIVE WORD, AND SENSORY PATTERN SEARCH RESULT B FOR CERTAIN ASSOCIATIVE WORD
S2145
S2146
INTEGRATED MATCHING LEVEL > THRESHOLD VALUE X?
NO
YES
HOLD AS SEARCH RESULT
S2147
YES
NEXT IMAGE?
S2148
NO
STORE SET OF IMAGE ID AND ITS MATCHING LEVEL IN SEARCH RESULT HOLDING UNIT
S2149
END

FIG. 22

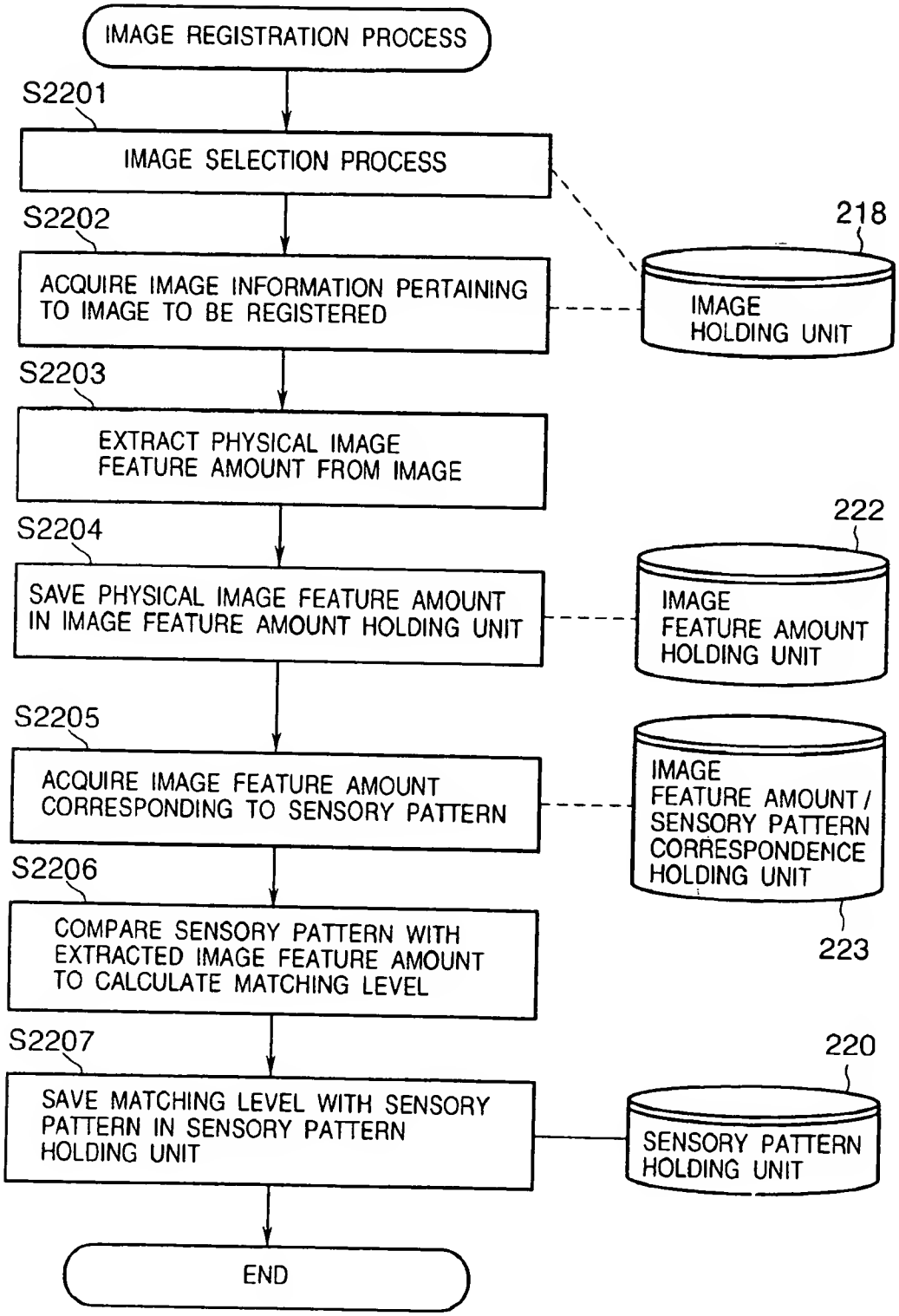


IMAGE REGISTRATION PROCESS
S2201
IMAGE SELECTION PROCESS
S2202
ACQUIRE IMAGE INFORMATION PERTAINING TO IMAGE TO BE REGISTERED
218
IMAGE HOLDING UNIT
S2203
EXTRACT PHYSICAL IMAGE FEATURE AMOUNT FROM IMAGE
S2204
SAVE PHYSICAL IMAGE FEATURE AMOUNT IN IMAGE FEATURE AMOUNT HOLDING UNIT
222
IMAGE FEATURE AMOUNT HOLDING UNIT
S2205
ACQUIRE IMAGE FEATURE AMOUNT CORRESPONDING TO SENSORY PATTERN
IMAGE FEATURE AMOUNT / SENSORY PATTERN CORRESPONDENCE HOLDING UNIT
223
S2206
COMPARE SENSORY PATTERN WITH EXTRACTED IMAGE FEATURE AMOUNT TO CALCULATE MATCHING LEVEL
S2207
SAVE MATCHING LEVEL WITH SENSORY PATTERN IN SENSORY PATTERN HOLDING UNIT
220
SENSORY PATTERN HOLDING UNIT
END

INFORMATION SEARCH APPARATUS AND METHOD, AND COMPUTER READABLE MEMORY

BACKGROUND OF THE INVENTION

The present invention relates to an information search apparatus and method for searching information on the basis of an input query word.

A conventional information search apparatus, which searches multimedia information, e.g., image information, makes a search using data (keywords) derived from subjective evaluation results of one or a plurality of persons for images to be searched, physical image features extracted from images, and the like.

Also, an image search apparatus that obtains a required image by matching a given keyword with that corresponding to an image has been realized. Furthermore, an information search apparatus, which obtains an image, that cannot be obtained by full-word matching with an input keyword, by matching not only the input keyword but also an associated keyword associated with the input keyword with a keyword corresponding to an image, has also been realized. Moreover, an information search apparatus which obtains an image with similar color information by detecting a correspondence between the input keyword and color information using, e.g., color information of images is proposed.

For example, in one scheme, an impression that a person receives upon watching an image, or key information linked with the impression is appended to image information and is used in search. As the key information, words that express impressions evoked by images such as "warm", "cold", and the like, and words that represent objects in drawn images such as "kitty", "sea", "mountain", and the like are appended as keywords. Also, local image feature components on drawn images are subjectively evaluated and are often appended as key information. For example, information that pertains to a color such as "red", "blue", and the like, information that pertains to a shape such as "round", "triangular, "sharp", and the like, and information that pertains to a texture such as "sandy", "smooth", and the like are expressed using words and icons, are appended to images as key information, and are used in search.

In another system, physical image feature amounts are extracted from images, and are used in image search. Image features include local colors painted on images, overall color tones, and shapes, compositions, textures, and the like of objects on drawn images. An image feature amount is extracted from segmented regions or blocks obtained by segmenting the overall image into regions based on color information, or segmenting the image into blocks each having a given area, or is extracted from the entire image. Physical image features include, e.g., color information, density distribution, texture, edge, region, area, position, frequency distribution, and the like of an image.

However, in the prior art, when an image including a keyword that matches the input query word is searched for, images which do not match the search request of the searcher are often obtained. Especially, when an image search is made using an abstract query word such as a "refreshing" image, images found by the search are limited. To solve this problem, a search may be made by unfolding the query word "refreshing" to keywords which are associated with that query word. However, when such scheme is used, images which are not "refreshing" may be mixed in search results.

In place of query words, a query image may be input, and a search may be made using the feature amount of the input image. In this case, a query image that reflects the searcher's will must be prepared, and it is difficult to select a query image, resulting in poor operability.

SUMMARY OF THE INVENTION

The present invention has been made in consideration of the above-mentioned problems, and has as its object to provide an image search method and apparatus which can extract desired information with high precision with respect to an input query word.

In order to achieve the above object, an image search apparatus according to the present invention comprises the following arrangement.

That is, there is provided an information search apparatus for searching information based on an input query word, comprising:

first search means for determining a query keyword on the basis of the query word, and searching information on the basis of the query keyword;

second search means for determining a feature amount of a pattern corresponding to the query word, and searching information on the basis of the feature amount; and

integration means for integrating search results obtained by the first and second search means.

In order to achieve the above object, an image search method according to the present invention comprises the following arrangement.

That is, there is provided an information search method for searching information based on an input query word, comprising:

the first search step of determining a query keyword on the basis of the query word, and searching information on the basis of the query keyword;

the second search step of determining a feature amount of a pattern corresponding to the query word, and searching information on the basis of the feature amount; and

the integration step of integrating search results obtained in the first and second search steps.

In order to achieve the above object, an image search apparatus according to the present invention comprises the following arrangement.

That is, there is provided an information search apparatus for managing a plurality of kinds of multimedia information, and searching the managed multimedia information for desired multimedia information, comprising:

a content word holding unit for storing the multimedia information, and content words which verbalize concepts expressed in the multimedia information in correspondence with each other;

an associative word dictionary for storing the content words and associative words which are associated with the content words in correspondence with each other;

input means for inputting a query word;

first search means for acquiring an associative word corresponding to the query word input by the input means from the associative word dictionary, and searching multimedia information on the basis of the acquired associative word;

extraction means for extracting a feature amount corresponding to the query word input by the input means;

second search means for searching multimedia information on the basis of the feature amount extracted by the extraction means; and

integration means for integrating search results obtained by the first and second search means.

In order to achieve the above object, an image search method according to the present invention comprises the following arrangement.

That is, there is provided an information search method for managing a plurality of kinds of multimedia information, and searching the managed multimedia information for desired multimedia information, comprising:

the storage step of storing on a storage medium a content word holding unit for storing the multimedia information, and content words which verbalize concepts expressed in the multimedia information in correspondence with each other, and an associative word dictionary for storing the content words and associative words which are associated with the content words in correspondence with each other;

the input step of inputting a query word;

the first search step of acquiring an associative word corresponding to the query word input in the input step from the associative word dictionary, and searching multimedia information on the basis of the acquired associative word;

the extraction step of extracting a feature amount corresponding to the query word input in the input step;

the second search step of searching multimedia information on the basis of the feature amount extracted in the extraction step; and

the integration step of integrating search results obtained in the first and second search steps.

In order to achieve the above object, a computer readable memory according to the present invention comprises the following arrangement.

That is, there is provided a computer readable memory for storing a program code of an information search process for managing a plurality of kinds of multimedia information, and searching the managed multimedia information for desired multimedia information, comprising:

a program code of the storage step of storing on a storage medium a content word holding unit for storing the multimedia information, and content words which verbalize concepts expressed in the multimedia information in correspondence with each other, and an associative word dictionary for storing the content words and associative words which are associated with the content words in correspondence with each other;

a program code of the input step of inputting a query word;

a program code of the first search step of acquiring an associative word corresponding to the query word input in the input step from the associative word dictionary, and searching multimedia information on the basis of the acquired associative word;

a program code of the extraction step of extracting a feature amount corresponding to the query word input in the input step;

a program code of the second search step of searching multimedia information on the basis of the feature amount extracted in the extraction step; and

a program code of the integration step of integrating search results obtained in the first and second search steps.

In order to achieve the above object, an image search apparatus according to the present invention comprises the following arrangement.

That is, there is provided an information search apparatus for managing a plurality of kinds of multimedia information, and searching the managed multimedia information for desired multimedia information, comprising:

a content word holding unit for storing the multimedia information, and content words which verbalize concepts expressed in the multimedia information in correspondence with each other;

an associative word dictionary for storing the content words and associative words which are associated with the content words in correspondence with each other;

input means for inputting a query word;

a concept discrimination dictionary for storing index words corresponding to the query word and search perspectives pertaining to the index words in correspondence with each other;

display means for extracting search perspectives pertaining to an index word corresponding to the query word input by the input means from the concept discrimination dictionary, and displaying the extracted search perspectives;

designation means for designating a desired one of the search perspectives displayed by the display means;

first search means for acquiring an associative word corresponding to the query word input by the input means from the associative word dictionary, and searching multimedia information on the basis of the acquired associative word;

second search means for extracting a feature amount corresponding to the query word input by the input means, and searching multimedia information on the basis of the extracted feature amount; and

integration means for integrating search results obtained by the first and second search means on the basis of the search perspective designated by the designation means.

In order to achieve the above object, an image search method according to the present invention comprises the following arrangement.

That is, there is provided an information search method for managing a plurality of kinds of multimedia information, and searching the managed multimedia information for desired multimedia information, comprising:

the input step of inputting a query word;

the storage step of storing on a storage medium a content word holding unit for storing the multimedia information, and content words which verbalize concepts expressed in the multimedia information in correspondence with each other, an associative word dictionary for storing the content words and associative words which are associated with the content words in correspondence with each other, and a concept discrimination dictionary for storing index words corresponding to the query word and search perspectives pertaining to the index words in correspondence with each other;

the display step of extracting search perspectives pertaining to an index word corresponding to the query word input in the input step from the concept discrimination dictionary, and displaying the extracted search perspectives;

the designation step of designating a desired one of the search perspectives displayed in the display step;

the first search step of acquiring an associative word corresponding to the query word input in the input step from the associative word dictionary, and searching multimedia information on the basis of the acquired associative word;

the second search step of extracting a feature amount corresponding to the query word input in the input step, and searching multimedia information on the basis of the extracted feature amount; and

the integration step of integrating search results obtained in the first and second search steps on the basis of the search perspective designated in the designation step.

In order to achieve the above object, a computer readable memory according to the present invention comprises the following arrangement.

That is, there is provided a computer readable memory for storing a program code of an information search process for managing a plurality of kinds of multimedia information, and searching the managed multimedia information for desired multimedia information, comprising:

a program code of the input step of inputting a query word;

a program code of the storage step of storing on a storage medium a content word holding unit for storing the multimedia information, and content words which verbalize concepts expressed in the multimedia information in correspondence with each other, an associative word dictionary for storing the content words and associative words which are associated with the content words in correspondence with each other, and a concept discrimination dictionary for storing index words corresponding to the query word and search perspectives pertaining to the index words in correspondence with each other;

a program code of the display step of extracting search perspectives pertaining to an index word corresponding to the query word input in the input step from the concept discrimination dictionary, and displaying the extracted search perspectives;

a program code of the designation step of designating a desired one of the search perspectives displayed in the display step;

a program code of the first search step of acquiring an associative word corresponding to the query word input in the input step from the associative word dictionary, and searching multimedia information on the basis of the acquired associative word;

a program code of the second search step of extracting a feature amount corresponding to the query word input in the input step, and searching multimedia information on the basis of the extracted feature amount; and

a program code of the integration step of integrating search results obtained in the first and second search steps on the basis of the search perspective designated in the designation step.

Other features and advantages of the present invention will be apparent from the following description taken in conjunction with the accompanying drawings, in which like reference characters designate the same or similar parts throughout the figures thereof.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a block diagram showing the arrangement of an information search apparatus according to an embodiment of the present invention;

FIG. **2** is a block diagram showing the functional arrangement of the information search apparatus according to the embodiment of the present invention;

FIG. **3** is a table showing the structure of a sensory pattern/associative word correspondence holding unit in the embodiment of the present invention;

FIG. **4** is a table showing the structure of an unfolded sensory pattern holding unit in the embodiment of the present invention;

FIG. **5** is a table showing the structure of a sensory pattern holding unit in the embodiment of the present invention;

FIG. **6** is a table showing an example of image feature amounts in the embodiment of the present invention;

FIG. **7** is a table showing the structure of an image feature amount & sensory pattern holding unit in the embodiment of the present invention;

FIG. **8** shows a display example of a search perspective input by a search request input processing unit in the embodiment of the present invention;

FIG. **9** shows a display example on a control panel upon instructing search weights in the embodiment of the present invention;

FIG. **10** is a table showing the structure of an image holding unit in the embodiment of the present invention;

FIG. **11** is a table showing the structure of an image content word holding unit in the embodiment of the present invention;

FIG. **12** is a table showing another example of the structure of an image content word holding unit in the embodiment of the present invention;

FIG. **13** is a table showing the structure of a concept discrimination dictionary in the embodiment of the present invention;

FIG. **14** is a table showing the structure of an associative word dictionary in the embodiment of the present invention;

FIG. **15** is a table showing the structure of a search result holding unit in the embodiment of the present invention;

FIG. **16** is a table showing another example of feature amounts in the embodiment of the present invention;

FIG. **17** is a table showing the structure of an image feature amount holding unit in the embodiment of the present invention;

FIG. **18** is a flow chart showing processes executed in the embodiment of the present invention;

FIG. **19** is a flow chart showing details of a search request input process in the embodiment of the present invention;

FIG. **20** is a flow chart showing details of a search process using an associative word in the embodiment of the present invention;

FIG. **21** is a flow chart showing details of a search process using a sensory pattern in step **S3006** and a search result combining process in step **S3007** in the embodiment of the present invention; and

FIG. **22** is a flow chart showing an image registration process in the embodiment of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

The preferred embodiments of the present invention will be described in detail hereinafter with reference to the accompanying drawings.

FIG. **1** shows the arrangement of an information search apparatus according to an embodiment of the present invention.

Referring to FIG. **1**, reference numeral **11** denotes a microprocessor (CPU), which makes computations, logical decisions, and the like for image information search in accordance with control programs, and controls individual building components connected to an address bus AB,

control bus CB, and data bus DB via these buses. The address bus AB transfers an address signal indicating the building component to be controlled by the CPU **11**. The control bus CB transfers and applies a control signal for each building component to be controlled by the CPU **11**. The data bus DB transfers data among the respective building components.

Reference numeral **12** denotes a read-only memory (ROM), which stores control programs such as a boot processing program executed by the CPU **11** upon starting up the apparatus of this embodiment, a processing program executed in this embodiment, and the like. Reference numeral **13** denotes a rewritable random access memory (RAM) which is configured by 16 bits per word, and is used as a temporary storage of various data from the respective building components. Also, the RAM **13** stores a query word holding unit **202**, search perspective holding unit **203**, search weight holding unit **204**, determined weight holding unit **207**, unfolded associative word holding unit **209**, unfolded sensory pattern holding unit **213**, and search result holding unit **216**, which will be described later with reference to FIG. **2**.

Reference numeral **14** denotes an external memory (DISK), which stores a concept discrimination dictionary **205**, associative word dictionary **211**, image word/sensory pattern correspondence holding unit **215**, image content word holding unit **219**, image holding unit **218**, sensory pattern holding unit **220**, image feature amount holding unit **222**, and image feature amount/sensory pattern correspondence holding unit **223**, which will be described later with reference to FIG. **2**. Also, the external memory **14** stores programs for respectively implementing processing units, i.e., a search request input processing unit **201**, weight determination processing unit **206**, associative word unfolding processing unit **208**, image content word search unit **210** using associative words, sensory pattern unfolding processing unit **212**, sensory pattern search processing unit **214**, search result integration processing unit **217**, image feature amount extraction processing unit **221**, and sensory pattern determination processing unit **224**, which will be described later with reference to FIG. **2**. As a storage medium for storing these programs, a ROM, floppy disk, CD-ROM, memory card, magnetooptical disk, or the like can be used.

Reference numeral **15** denotes a keyboard (KB) which has alphabet keys, hiragana keys, katakana keys, character symbol input keys for inputting a period, comma, and the like, a search key for instructing a search (a function key on a general keyboard may be used instead), and various function keys such as cursor moving keys for instructing cursor movement, and the like. Also, a pointing device such as a mouse or the like (not shown) may be connected.

Reference numeral **16** denotes a display video memory (VRAM) for storing a pattern of data to be displayed. Reference numeral **17** denotes a CRT controller (CRTC) for displaying the contents stored in the VRAM **16** on a CRT **18**. Reference numeral **18** denotes a display device (CRT) using, e.g., a cathode ray tube, or the like. The dot display pattern and cursor display on the CRT **18** are controlled by the CRTC **17**. Note that various other displays such as a liquid crystal display, and the like may be used as the display device. Reference numeral **19** denotes a network controller (NIC), which connects the apparatus to a network such as Ethernet or the like.

The information search apparatus constructed by the aforementioned building components operates in accordance with various inputs from the KB **15** and various inputs supplied from the NIC **19** via the network. Upon receiving the input from the KB **15** or NIC **19**, an interrupt signal is sent to the CPU **11**. Upon receiving the interrupt signal, the CPU **11** reads out various control signals stored in the DISK **14**, and executes various kinds of control in accordance with these control signals. Also, the present invention is achieved by supplying a storage medium that stores a program according to the present invention to a system or apparatus, and by reading out and executing program codes stored in the storage medium by a computer of the system or apparatus.

The functional arrangement of the information search apparatus of this embodiment will be explained below with reference to FIG. **2**.

FIG. **2** is a block diagram showing the functional arrangement of the information search apparatus according to the embodiment of the present invention.

Referring to FIG. **2**, reference numeral **201** denotes a search request input processing unit for inputting query items (query word, search perspective or category, search weight, and the like) that pertain to the information wanted. Reference numeral **202** denotes a query word holding unit for storing a query word input by the search request input processing unit **201**. Reference numeral **203** denotes a search perspective holding unit for storing a search perspective input by the search request input processing unit **201**. Reference numeral **204** denotes a search weight holding unit for storing a search weight input by the search request input processing unit **201**.

Reference numeral **205** denotes a concept discrimination dictionary having a search perspective that pertains to a concept as the information wanted, an antithetic concept having a contrary or antonymous meaning, and two kinds of coefficients for weight discrimination upon searching for a concept. Reference numeral **206** denotes a weight determination processing unit for giving weights (associated weight and sensory pattern weight) indicating the weight balance on associative words (obtained by an associative word unfolding processing unit **208**) and sensory patterns (obtained by a sensory pattern unfolding processing unit **212**) upon searching using a query word stored in the query word holding unit **202**. Reference numeral **207** denotes a determined weight holding unit for holding the search weight determined by the weight determination processing unit **206**.

Reference numeral **208** denotes an associative word unfolding processing unit for unfolding the query word stored in the query word holding unit **202** into associative words with reference to an associative word dictionary **211**, obtaining an antithetic concept antonymous to that query word from the concept discrimination dictionary **205**, and unfolding the antithetic concept into associative words with reference to the associative word dictionary **211**. Reference numeral **209** denotes an unfolded associative word holding unit for holding the associative words (including those of the antithetic concept) unfolded by the associative word unfolding processing unit **208**. Reference numeral **210** denotes an image content word search processing unit using associative words, which finds image content words, which are stored in an image content word holding unit **219** and match the unfolded associative words, by search with reference to the unfolded associative word holding unit **209**.

Reference numeral **211** denotes an associative word dictionary for storing associative words to be unfolded in units of concepts serving as index words in correspondence with associative perspectives. Reference numeral **212** denotes a sensory pattern unfolding processing unit for unfolding the

query word stored in the query word holding unit **202** into sensory patterns with reference to an image word/sensory pattern correspondence holding unit **215**, obtaining an antithetic concept antonymous to the stored query word from the concept discrimination dictionary **205**, and unfolding the obtained antithetic concept into sensory patterns with reference to the image word/sensory pattern correspondence holding unit **215**.

Reference numeral **215** denotes an image word/sensory pattern correspondence holding unit for storing image words and sensory patterns in correspondence with each other, i.e., storing image words and sensory pattern IDs corresponding to associative words, which are associated with the image words. Note that FIG. **3** shows a data storage example of the image word/sensory pattern correspondence holding unit **215**. The structure of the image word/sensory pattern correspondence holding unit **215** will be described in detail later.

Reference numeral **213** denotes an unfolded sensory pattern holding unit for temporarily storing the sensory patterns unfolded by the sensory pattern unfolding processing unit **212**. The unit **213** is stored in the RAM **13**. Note that FIG. **4** shows a data storage example of the sensory pattern unfolding processing unit **213**. The structure of the sensory pattern unfolding processing unit **213** will be described in detail later.

Reference numeral **214** denotes a sensory pattern search processing unit for finding sensory patterns, which are stored in the sensory pattern holding unit **220** and are similar to the unfolded sensory patterns, by search with reference to the sensory pattern holding unit **220**. Reference numeral **217** denotes a search result integration processing unit for integrating the search results of image content words using the associative words, and the search results of sensory patterns stored in a search result holding unit **216**, on the basis of the search weights obtained by the weight determination processing unit **206**.

Reference numeral **219** denotes an image content word holding unit for verbalizing and storing concepts expressed in image information stored in an image holding unit **218**. Reference numeral **218** denotes an image holding unit for storing image information serving as test images. Reference numeral **220** denotes a sensory pattern holding unit for holding sensory patterns obtained from the image information stored in the image holding unit **218**, and storing matching levels with respective sensory patterns in units of image IDs each indicating image information. Note that FIG. **5** shows a data storage example of the sensory holding unit **220**. The structure of the sensory pattern holding unit **220** will be described in detail later.

Reference numeral **221** denotes an image feature extraction processing unit for extracting physical image feature amounts from image information stored in the image holding unit **218**. Physical image feature amounts are visual features or signatures extracted from regions segmented on the basis of color information, blocks each segmented to have a given area, or the entire image. The image feature amount is, e.g., numerical information such as the color distribution or histogram, density distribution, texture, edge, frequency distribution, and the like of an image, and is expressed, as shown in, e.g., FIG. **6**. Note that the image feature amounts will be described in detail later.

Reference numeral **222** denotes an image feature amount holding unit for storing the image feature amounts obtained by the image feature amount extraction processing unit **221**. Reference numeral **223** denotes an image feature amount/sensory pattern correspondence holding unit for storing image feature amounts and sensory patterns in correspondence with each other, i.e., storing sensory pattern IDs and image feature amounts corresponding to those IDs. Note that FIG. **7** shows a data storage example of the image feature amount/sensory pattern correspondence holding unit **223**. The image feature amount/sensory pattern correspondence holding unit **223** will be described in detail later.

Reference numeral **224** denotes a sensory pattern determination processing unit for comparing a sensory pattern and image feature amount extracted from image information to obtain their matching level with reference to the image feature amount/sensory pattern correspondence holding unit **223**, and registering the matching level in the sensory pattern holding unit **220**.

A display example of a search perspective that pertains to search request items input at the search request input processing unit **201** will be explained below with reference to FIG. **8**.

FIG. **8** shows a display example of a search perspective input at the search request input processing unit in the embodiment of the present invention.

When a query word is input by operating, e.g., the keyboard **15**, the concept discrimination dictionary **205** shown in FIG. **13** is searched using the query word as an index word to extract corresponding search perspectives.

FIG. **8** illustrates that three search perspectives "color tone", "taste", and "general atmosphere" are available in relation to a query word "mild", and hatched "color tone" is selected as the search perspective. When the user presses an OK button in this state, the search perspective "color tone" is selected, and is held in the search perspective holding unit **203**. Also, the query word "mild" is held in the query word holding unit **202**.

By pressing one of the cursor moving keys on the keyboard **15**, the hatching moves from "color tone" to "taste" or "general atmosphere", and the user can designate a desired search perspective or category.

A display example on the control panel when the operator instructs the search weight balance on a search using associative words and a search using sensory patterns in actual search will be explained below with reference to FIG. **9**. As described above, a search using associative words and a search using the feature amounts of images (sensory patterns) based on the query word are made, and the search results are integrated. In this integration process, the two search results are weighted. On this control panel, the user can designate a search weight for a search using associative words, and that for a search using sensory patterns. That is, the user can designate the weight balance on a search using associative words and that using sensory patterns in actual search.

FIG. **9** shows a display example of the control panel upon instructing search weights in the embodiment of the present invention.

Referring to FIG. **9**, when the user slides a slide button **41** to the left, an instruction that sets a heavier weight on a search using associative words is issued; when he or she slides the slide button **41** to the right, an instruction that sets a heavier weight on a search using sensory patterns is issued. When the user designates search weights using the slide button **41** and then presses an OK button **43**, a search weight instruction is issued. A button **42** in the display area is pressed when no search weights are clearly designated, and in such case, a predetermined search weight instruction is issued. Upon depression of the button **42**, predetermined

weight values (which are obtained from an associated weight **83** and sensory pattern weight **84** in the concept discrimination dictionary **205**) are used. The set weights are stored in the search weight holding unit **204**. Note that the buttons **41** to **43** on the control panel may he clicked by a pointing device (not shown).

The structure of the image holding unit **218** will he described below using FIG. **10**.

FIG. **10** shows the structure of the image holding unit in the embodiment of the present invention.

The image holding unit **218** manages image information hy storing image IDs each indicating image information (image files) and image file storage paths each indicating the storage location of image information. Referring to FIG. **10**, reference numeral **2180** denotes an image ID which is uniquely assigned to one image file. Reference numeral **2181** denotes a file path which indicates the storage location of an image file corresponding to the image ID in the DISK **14**, and corresponds to the directory and file of MS-DOS.

An image file is divided into header and image data fields (not shown in FIG. **10**). The header field stores information required for reading data from that image file, and additional information that explains the image contents. As such information, an image format identifier indicating the image format name of the image, file size, image width, height, and depth, the presence/absence of compression, color pallet information, resolution, offset to the storage location of image data, and the like are stored. The image data field stores image data in turn. This embodiment uses the BMP format of Microsoft Corp. as such image format, hut other compression formats such as GIF, JPEG, FlashPix, and the like may be used.

The structure of the image content word holding unit **219** will be described below with the aid of FIG. **11**.

FIG. **11** shows the structure of the image content word holding unit in the embodiment of the present invention.

The image content word holding unit **219** manages image information by storing the image IDs and image content words in correspondence with each other. Referring to FIG. **11**, reference numeral **21900** denotes a field for storing image IDs corresponding to the image IDs **2180** shown in FIG. **10**; and **21901**, a field for storing image content words that express image files corresponding to the image IDs **21900**. The image content word verhalizes an image feature expressed in an image file, and stores a keyword as a character code (e.g., unicode). A plurality of keywords may be stored per image file, and the image content word holding unit **219** is expressed as a list of image content words **21901** using image IDs **21900** as keys. Or, as shown in FIG. **12**, the image content word holding unit **219** may he expressed as a list of image IDs **21911** using image content words **21910** as keys.

FIG. **12** shows a tahle which stores data of the image content word holding unit **219** shown in FIG. **11** as a list of image IDs using image content words as keys. Referring to FIG. **12**, all image IDs **21911** that contain the individual words of image content words **21910** as keywords are stored. Note that FIG. **11** shows classification hased on image IDs, and FIG. **12** shows classification hased on image content words. Therefore, since FIGS. **11** and **12** have the same contents, both the tables need not always be held.

The structure of the concept discrimination dictionary **205** will he described helow using FIG. **13**.

FIG. **13** shows the structure of the concept discrimination dictionary in the embodiment of the present invention.

As shown in FIG. **13**, the concept discrimination dictionary **205** provides information that pertains to a query word serving as a search request, and stores index words **2050** corresponding to query words, search perspectives **2051** associated with index words **2050**, antithetic concepts **2052** having meanings contrary to the index words **2050**, associated weights **2053** used upon searching the index words **2050**, and sensory pattern weights **2054** used upon searching the index words **2050** in correspondence with each other.

The structure of the associative word dictionary **211** will he explained below with reference to FIG. **14**.

FIG. **14** shows the structure of the associative word dictionary in the emhodiment of the present invention.

The associative word dictionary **211** is composed of associative IDs **2110** each of which assigns a unique number to a set of associative words for each index word **2111**, index words **2111** each serving as a start point of association, associative words **2112** evoked hy the index words **2111**, associative perspectives **2113** which are relevant to associations of the associative words **2112**, and association strengths **2114** each indicating the strength of association hetween each pair of index word **2111** and associative word **2112**.

The association strength **2114** assumes an absolute value ranging from 0 to 10, and its sign indicates direction of association of the associative word. More specifically, when the association strength is a positive value, it indicates a stronger associative relationship (higher bilateral association) as the association strength value is larger; when the association strength is a negative value, it indicates a harder associative relationship as the association strength value is larger. For example, an associative word "folkcraft article" corresponding to an index word "simple" in associative data with the associative ID=126533 can be associated with strength "**6**", hut an associative word "chandelier" in associative data with the associative ID=126536 is hardly associated with strength "9" since its association strength is a negative value.

The structure of the search result holding unit **216** will he described below with reference to FIG. **15**.

FIG. **15** shows the structure of the search result holding unit in the embodiment of the present invention.

The search result holding unit **216** stores image IDs which are found hy searches of the image content word search processing unit **210** using associative words and the sensory pattern search processing unit **214**. Referring to FIG. **15**, reference numeral **2160** denotes a field for storing image IDs found by search; **2161**, a field for storing the number of matched associative words with positive association strengths by the image content word search processing unit **217** using associative words; and **2162**, a field for storing a list a maximum of **20** associative word IDs **2110** of matched associative words in the associative word dictionary **211**. When the number **2161** of matched associative words is zero, the associative ID **2162** is filled with NULL code. Reference numeral **2163** denotes a field for storing the search matching levels of associative words with respect to the image IDs **2160**. When the numher **2161** of matched associative words is zero, the associative matching level **2163** stores zero.

Reference numeral **2164** denotes a field for storing the number of sensory patterns with highest similarity, which are found by search hy the sensory pattern search processing unit **223**; and **2165**, a field for storing a list of a maximum of 20 sensory pattern IDs of matched sensory patterns. When the numher **2164** of matched sensory patterns is zero, the

sensory pattern ID **2165** is filled with NULL code. Reference numeral **2166** denotes a field for storing the search matching level of a sensory pattern search with respect to the image ID **2160**. When the number **2164** of matched sensory patterns is zero, the sensory pattern matching level **2166** stores zero. Reference numeral **2167** denotes a field for storing the matching level (obtained by the search result integration processing unit **217**) of the image ID **2160** with respect to the search request, which is calculated using the associative matching level **2163** and sensory pattern matching level **2166** as parameters.

The structure of the above-mentioned unfolded sensory pattern holding unit **213** will be described below with reference to FIG. **4**.

Referring to FIG. **4**, reference numeral **2130-1** denotes an image word as an unfolding source from which this sensory pattern has evolved upon unfolding, and the same image word as that in the query word holding unit **202** is stored. In this embodiment, a character string "refreshing" is stored, and ends with NULL code. Reference numeral **2130-2** denotes the number of sensory patterns obtained by unfolding the image word **2130-1** with reference to the image word/sensory pattern correspondence holding unit **215**. For example, when the contents of the image word/sensory pattern correspondence holding unit **215** are as shown in FIG. **4**, the number of sensory patterns unfolded from the image word "refreshing" is 7. Reference numeral **2130-3** denotes an address indicating the storage location area of data obtained by actually unfolding the image word "refreshing". The address **2130-3** is linked with unfolded data **2130-4**.

Reference numeral **2130-4** denotes unfolded data actually unfolded from "refreshing", and sets of associative words and sensory patterns corresponding to the number **2130-2** of sensory patterns are stored here. In this embodiment, seven sets of associative words and sensory patterns are stored. Reference numeral **2130-5** denotes a sensory pattern ID corresponding to the image word "refreshing" and an associative word "forest" of "refreshing". In this embodiment, "5" is stored. Reference numeral **2130-6** denotes an associative word of the image word "refreshing". In this embodiment, a character string "forest" is stored, and ends with NULL code.

The structure of the aforementioned image word/sensory pattern correspondence holding unit **215** will be described in detail below using FIG. **3**.

Referring to FIG. **3**, reference numeral **2150-1** denotes an image word serving as an unfolding source of this sensory pattern. In this embodiment, character strings "refreshing", "tropical", and the like are stored, and end with NULL code. Reference numeral **2150-2** denotes an associative word unfolded from the image word **2150-1**. In this embodiment, associative words "forest", "tableland", "blue sky", and the like are stored in correspondence with "refreshing", and these character strings end with NULL code. When no character string is stored in this field, i.e., NULL code alone is stored, this sensory pattern applies to all image words "refreshing"; no specific associative word has been designated.

Reference numeral **2150-3** denotes a sensory pattern ID corresponding to the image word **2150-1** and associative word **2150-2**. In this embodiment, "005" and "006" are stored as sensory pattern IDs corresponding to the image word "refreshing" and its associative word "forest". Also, sensory patterns for "not refreshing" as an antithetic concept of "refreshing" are stored. In this embodiment, for "not refreshing", no associative words are registered and "001" and "010" are registered as sensory pattern IDs.

The structure of the above-mentioned sensory pattern holding unit **220** will be described in detail below using FIG. **5**.

Referring to FIG. **5**, reference numeral **2200-1** denotes an image ID for identifying an image to be registered. The image IDs use the same ones as those stored in the image holding unit **218**, and uniquely define images in this system. A field **2200-2** stores sensory pattern IDs. In this embodiment, since the matching levels between each image and all sensory patterns stored in the image feature amount/sensory pattern correspondence holding unit **223** are calculated, all the sensory pattern IDs are stored. Reference numeral **2200-3** denotes a numerical value indicating the matching level between each image and sensory pattern. The matching level assumes a value ranging from 0 to 1; 0 indicates the image does not match the sensory pattern at all, and the matching level becomes higher as it is closer to 1. For example, the matching level between image with the image ID=001 and sensory pattern **1** is 0.10, and the matching level between that image and sensory pattern **2** is **0**.

The aforementioned image feature amounts will be explained in detail below with reference to FIG. **6**. In FIG. **6**, X1, X2, X3, . . . , Xn represent image features extracted from one image, B1, B2, . . . , Bm represent regions/blocks from which image feature amounts are extracted, and x11 to xmn represent image feature amounts extracted from the individual regions/blocks. That is, feature amounts that pertain to physical image features X1 to Xn are obtained in units of regions/blocks. FIG. **16** exemplifies a case wherein chromatic image feature amounts are extracted, i.e., representative colors are extracted in units of image regions or blocks. Referring to FIG. **16**, a physical image feature is expressed by "representative color" and "feature amount", representative colors extracted from regions/blocks B1, B2, . . . , Bn are C1 (R1, G1, B1), C2(R2, G2, B2), . . . , Cn(Rn, Gn, Bn), and their image feature amounts are c1 to cn.

The structure of the image feature amount holding unit **222** will be described below using FIG. **17**.

FIG. **17** shows the structure of the image feature amount holding unit in the embodiment of the present invention.

Referring to FIG. **17**, reference numeral **2220-1** denotes an image ID for identifying an image to be registered. The image IDs use the same ones as those stored in the image holding unit **218**. Reference numeral **2220-2** denotes a block or region number from which an image feature amount is extracted. In this embodiment, B1, B2, . . . , Bm represent the region/block numbers. Reference numeral **2220-3** denotes information (in this embodiment, a representative color is used) indicating an image feature extracted from each of the regions/blocks B1, B2, . . . , Bm (**2220-2**). This embodiment exemplifies a case wherein chromatic image features are extracted, and a plurality of pieces of information C11(R11, G11, B11), . . . , Cn1(Rn1, Gn1, Bn1) indicating colors are stored. Reference numeral **2220-4** denotes image feature amounts of image features extracted from the individual regions/blocks. In this embodiment, c11, . . . , cn1 are stored as the image feature amounts of image features C11(R11, G11, B11), . . . , Cn1(Rn1, Gn1, Bn1).

The structure of the image feature amount/sensory pattern correspondence holding unit **223** will be described in detail below using FIG. **7**.

Referring to FIG. **7**, reference numeral **2230-1** denotes a sensory pattern ID, which uniquely identifies a sensory

pattern. Reference numeral **2230-2** denotes an image feature amount corresponding to each sensory pattern ID. In this embodiment, a sensory pattern is expressed by a chromatic image feature amount, and a combination of color components (values in a color space such as RGB, HVC, or the like) corresponding to each sensory pattern ID is stored. In this embodiment, values in the RGB color space are registered as color components. The RGB values assume integers ranging from **0** to 255, and a maximum of m colors correspond to each sensory pattern ID.

The sensory pattern determination processing unit **224** calculates the matching levels between each of image data registered in the image holding unit **218** and the respective sensory patterns using the aforementioned image feature amount holding unit **222** and image feature amount/sensory pattern correspondence holding unit **223**, and registers them in the sensory pattern holding unit **220** (to be described later in step S2207 in FIG. **22**).

The processes executed in this embodiment will be described below using FIG. **18**.

FIG. **18** is a flow chart showing the processes executed in the embodiment of the present invention.

In step **S3001**, a processing module that implements the operation of the search request input processing unit **201** in FIG. **2** executes a search request input process. Note that the search request input process will be explained in detail later. If it is determined with reference to the contents of the search weight holding unit **204** in step **S3002** that search weights are designated, the designated values are stored in the determined weight holding unit **207**. On the other hand, if no search weights are designated, index words **2050** are searched for a query word stored in the query word holding unit **202** with reference to the concept discrimination dictionary **205** so as to read out a corresponding associated weight **2053** and sensory pattern weight **2054**, and the readout weights are stored in the determined weight holding unit **207**. If there is no index word **2050** that is relevant to the contents of the query word holding unit **202**, a default value "5" is stored as both the associated and sensory pattern weights in the determined weight holding unit **207**.

It is checked with reference to the determined weight holding unit **207** in step **S3003** if the associated weight is zero. If the associated weight is zero (YES in step **S3003**), the flow advances to step **S3005**. On the other hand, if the associated weight is not zero (NO in step **S3003**), the flow advances to step **S3004**.

In step **S3004**, a processing module that implements the operations of the associative word unfolding processing unit **208** and image content word search processing unit **210** using associative words in FIG. **2** executes a search process using associative words. Note that the search process using associative words will be described in detail later.

It is checked with reference to the determined weight holding unit **207** in step **S3005** if the sensory pattern weight is zero. If the sensory pattern weight is zero (YES in step **S3005**), the flow advances to step **S3007**. On the other hand, if the sensory pattern weight is not zero (NO in step **S3005**), the flow advances to step **S3006**.

In step **S3006**, a processing module that implements the operations of the sensory pattern unfolding processing unit **212** and sensory pattern search processing unit **214** in FIG. **2** executes a search process using sensory patterns. Note that the search process using sensory patterns will be described in detail later. In step **S3007**, a processing module that implements the operation of the search result integration processing unit **217** executes a search result integration process. Note that the search result integration process will be described in detail later.

In step **S3008**, image files corresponding to image IDs stored in the search result holding unit **216** as search results obtained in step **S3007** are read out from the image holding unit **218**, and are displayed. Note that this process is a known one which is prevalent in image search apparatuses of the same type.

The search request input process in step **S3001** will be described in detail below with reference to FIG. **19**.

FIG. **19** is a flow chart showing the details of the search request input process in the embodiment of the present invention.

In step **S2011**, a query word serving as a search request is input. The query word input is attained by storing a character code input at the KB **15** in the query word holding unit **202** on the RAM **13**. In step **S2012**, search perspectives that are relevant to the query word stored in the query word holding unit **202** are extracted from the concept discrimination dictionary **205**. That is, all search perspectives **2051** corresponding to index words **2050**, which match the query word in the query word holding unit **202**, are extracted. For example, when the query word is "mild", three search perspectives "color tone", "taste", and "general atmosphere" can be obtained.

It is checked in step **S2013** if a search perspective or perspectives is or are found. If a search perspective or perspectives is or are found (YES in step **S2013**), the flow advances to step **S2014**. On the other hand, if no search perspective is found (NO in step S2013), the flow advances to step **S2016**.

In step **S2014**, the window for designating the search perspective described above with reference to FIG. **8** is displayed. In step **S2015**, the user selects a desired one of the search perspectives displayed on the window. The selected search perspective is stored in the search perspective holding unit **203**.

In step **S2016**, the user inputs search weights which determine the weight balance on a search using associative words and a search using sensory pattern in actual search in relation to the search process in response to the search request. That is, the user operates the slide button **41** on the control panel shown in FIG. **9** to designate the weight ratios on associative words and sensory patterns. When the user does not designate any search weights, he or she presses the button **42** in the display area on the control panel shown in FIG. **9** to designate default values of the search weights.

It is checked in step **S2017** if search weights are designated. If search weights are not designated (NO in step **S2017**), i.e., if the default values of the search weights are designated, the processing ends. On the other hand, if search weights are designated (YES in step **S2017**), the designated associative word and sensory pattern weights are stored in the search weight holding unit **204** in step **S2018**, thus ending the processing.

The search process using associative words in step **S3004** will be described in detail below with reference to the flow chart in FIG. **20**.

FIG. **20** is a flow chart showing the details of the search process using associative words in the embodiment of the present invention.

In step **S2101**, associative word data corresponding to index words **2111** in the associative word dictionary **211**, that match the query word stored in the query word holding unit **202**, are found by search. That is, the associative word

dictionary **211** is searched for index words **2150-2** (FIG. **3**), which match the query word, and registered associative word data are extracted. If index words that match the query word are found, all their associative IDs are stored in the unfolded associative word holding unit **209**.

In step **S2102**, the concept discrimination dictionary **205** is searched, and if an index word that matches the query word in the query word holding unit **202** is found, a search perspective **2051** corresponding to that index word is extracted. The extracted search perspective **2051** is compared with that stored in the search perspective holding unit **203**, and if they match, an antithetic concept **2052** corresponding to this index word is extracted. On the other hand, if the two search perspectives do not match, data in which the query word matches an index word continues to be searched for, and if no antithetic concept whose search perspective matches the index word is found finally, the flow advances to step **S2103**.

In step **S2103**, the associative word dictionary **211** is searched for associative words having an index word, which matches the antithetic concept found in step **S2102**. If an index word that matches the antithetic concept is found, their associative IDs are stored in the unfolded associative word holding unit **209** by appending a status code indicating an antithetic concept thereto.

In step **S2104**, associative words are extracted based on the associative IDs stored in the unfolded associative word holding unit **209**, and the image content word holding unit **219** is searched for image content words that match the associative words. The search results are stored in the search result holding unit **216**. More specifically, the associative IDs are extracted from the unfolded associative word holding unit **209**, and corresponding associative data are extracted with reference to the associative word dictionary **211**. Next, the association strengths **2114** of the extracted associative data are extracted, and if a status code indicating an antithetic concept is appended to a given associative ID extracted from the unfolded associative word holding unit **209**, the sign of the association strength is inverted to indicate a negative association strength. However, if the association strength is already a negative value, that associative data is discarded, and the next associative data is checked. In this manner, the obtained association strengths are set in a work memory ASCF (not shown) on the RAM **13**.

Then, an associative perspective corresponding to each associative ID is extracted, and is compared with that stored in the search perspective holding unit **203**. If the two perspectives match, a predetermined value a is set in a work memory VPF (not shown) on the RAM **13**. If they do not match, a value $\alpha \times 0.1$ is set in the work memory VPF on the RAM **13**.

Finally, the image content word holding unit **219** is searched for image content words that match associative words corresponding to the associative IDs. If an image content word is found, its image ID **21911** is set in the found image ID **2160** in the search result holding unit **216**, "1" is set in the number **2161** of matched associative words, and the found associative ID is set in the associative word ID **2162**. Then, a value obtained by multiplying the value in the work memories ASCF and VPF on the RAM **13** by a predetermined score β based on associative word matching is stored as an associative matching level in the associative matching level **2163**. If an identical image ID has already been stored, the value of the number **2161** of matched associative words is incremented by 1, a new associative word ID is added to the associative word ID **2162**, and the calculated associative matching level is added to the stored associative matching level **2163** to update its value.

The search process using sensory patterns in step **S3006** and the search result integration process in step **S3007** will be described in detail below with reference to FIG. **21**.

FIG. **21** is a flow chart showing the search process using sensory patterns in step **S3006** and the search result integration process in step **S3007** in the embodiment of the present invention.

These processes are controlled in accordance with a processing program stored in the DISK **14**.

The user inputs a search request for searching images at the search request input processing unit **201**. The search request contains one or a plurality of query words, search perspectives, and the like. The query word input in this embodiment is an abstract image word that expresses impressions of images such as "refreshing", "swarm", and the like. In this embodiment, assume that an image word "refreshing" is stored.

Steps **S2141** and **S2142** are implemented by the sensory pattern unfolding processing unit **212**. In step **S2141**, the image word held in the query word holding unit **202** is unfolded into sensory patterns with reference to the image word/sensory pattern correspondence holding unit **215**. In this embodiment, the query word holding unit **202** stores the image word "refreshing", the unfolded associative word holding unit **209** holds associative words "forest", "tableland", "blue sky", and the like unfolded from "refreshing", and the image word is unfolded into corresponding sensory pattern IDs with reference to the image word/sensory pattern correspondence holding unit **215**. For example, sensory pattern IDs "005" and "006", corresponding to image word "refreshing"—associative word "forest" are acquired, and a sensory pattern ID "007" corresponding to image word "refreshing"—associative word "tableland" is acquired.

In step **S2142**, the sets of unfolded sensory pattern IDs and image words/associative words are stored in the unfolded sensory pattern holding unit **213**. The data storage example at that time is as shown in FIG. **4**.

Steps **S2143** and **S2144** are implemented by the sensory pattern search processing unit **214**. In step **S2143**, all image IDs of images having matching levels larger than zero with respect to the sensory pattern IDs stored in the unfolded sensory pattern holding unit **213** are acquired. This process is done for all the sensory patterns stored in the unfolded sensory pattern holding unit **213**. Note that the sensory pattern search processing unit **214** acquires image IDs having matching levels larger than zero with respect to the sensory pattern IDs respectively unfolded from the query word and antithetic concept.

In step **S2144**, sets of acquired sensory pattern IDs, image IDs, and their matching levels are stored in the search result holding unit **216**.

Steps **S2145** to **S2149** are implemented by the search result integration processing unit **217**. In step **S2145**, two sets of search results, i.e., the image content word search results using associative words and sensory pattern search results, which are stored in the search result holding unit **216**, are integrated into one set of search results on the basis of the search weights stored in the determined weight holding unit **207** with reference to those search results. When the sensory pattern search results include a sensory pattern based on the antithetic concept to the query word, the corresponding image is excluded from the integrated results.

Or the sensory pattern matching level of an image including a sensory pattern of the antithetic concept may be lowered upon integration. In this process, a method of obtaining common elements of two sets of search results in units of associative words (ANDing search results), a method of calculating integrated matching levels based on the weights on the searches, and selecting appropriate search results in descending order of integrated matching levels, and the like are available. In this embodiment, the method of calculating the integrated matching levels will be exemplified below.

Let A be the associative matching level of an image that matches an associative word "forest" stored in the search result holding unit **216**, B be the sensory matching level of an image that matches the sensory pattern ID "005" corresponding to the associative word "forest", and w1 and w2 (w1+w2=1) be the search weights stored in the determined weight holding unit **207**. Then, the integrated matching level is given by:

$$\text{Integrated matching level}=w1*A+w2*B$$

or

$$\text{Integrated matching level}=(w1*A^2+w2*B^2)^{1/2}$$

The integrated matching levels of all sensory patterns of all associative words are calculated. When one image ID has matching levels larger than zero with respect to a plurality of sensory pattern IDs, a plurality of integrated matching levels are obtained for one image. However, in this case, an image with the highest integrated matching level is adopted as a search result. This process is done for all images corresponding to either set of search results larger than zero, and images whose integrated matching levels are larger than a predetermined threshold value X are selected as integrated search results.

That is, it is checked in step **S2146** if the integrated matching level of an image to be processed is larger than the threshold value X. If the integrated matching level is equal to or smaller than the threshold value X (NO in step **S2146**), the flow returns to step **S2145**. On the other hand, if the integrated matching level is larger than the threshold value X (YES in step **S2146**), the flow advances to step **S2147**. In step **S2147**, the image ID of the image to be processed is held in the search result holding unit **216** as a search result. It is checked in step **S2148** if the next image to be processed still remains. If the next image still remains (YES in step **S2148**), the flow returns to step **S2145**. On the other hand, if no images remain (NO in step **S2148**), the flow advances to step **S2149**.

In step **S2149**, the sets of image IDs and their integrated matching levels are stored in the search result holding unit **216**, thus ending the processing.

An image registration process for registering test images will be explained below with reference to FIG. **22**.

FIG. **22** is a flow chart showing the image registration process in the embodiment of the present invention.

This process is controlled in accordance with a processing program stored in the DISK **14**.

In step **S2201**, the user designates an image to be registered. The image to be registered is designated from those stored in an external storage device, an image input device, an image database server connected to this image processing apparatus, or the like (none of them are shown). In this embodiment, assume that images serving as test images are stored in advance, and the image to be registered is selected from them.

In step **S2202**, an image ID corresponding to an image file name of the designated image, and various kinds of image information required for registration are acquired, and are supplied to the image feature extraction processing unit **221**. The image ID is stored in correspondence with the image file name to manage an image, and is acquired by searching data in the image holding unit **218** using the image file name. Various kinds of image information of the image include pixel values indicating the width and height of an image, the number of bits per pixel, the image size (in units of bytes), the address of the area where a bitmap image is actually stored, and the like, for example, when the file format of this image is the bitmap format. Since these pieces of image information are stored in the header field of the image file, they can be acquired by referring to the header field. Even when the file format of the image is not the bitmap format but JFIF or FlashPix, required information can be similarly obtained from the header field of a file. Or the image holding unit **218** may store such image information, and the image information may be acquired by referring to the image holding unit **218** upon registration.

In step **S2203**, physical image feature amounts are extracted by analyzing the image information corresponding to the designated image ID. This process is done by the image feature amount extraction processing unit **221**. An example of this process is as has already been described previously with reference to FIG. **16**. FIG. **16** shows an example of the image feature amounts in this embodiment, and representative colors are extracted in units of image regions/blocks. The representative color may be obtained by using a scheme of analyzing an actual bitmap image using various kinds of input image information in units of pixels, and calculating the average value of color components (values in a color space such as RGB, HVC, or the like) used in each region or block, or a color component with the highest frequency of occurrence as a representative color.

In step **S2204**, extracted image feature amounts c1 to cn are stored in the image feature amount holding unit **222** in correspondence with the image ID of that image. This example is as has already been described previously with reference to FIG. **17**.

In step **S2205**, all sensory pattern IDs stored in the image feature amount/sensory pattern correspondence holding unit **223**, and image feature amounts corresponding to those sensory patterns are acquired with reference to the image feature amount/sensory pattern correspondence holding unit **223**. This example is as has already been described previously with reference to FIG. **7**.

In step **S2206**, the matching level between the acquired sensory pattern and the image feature amounts corresponding to the image is calculated. This process is done by the sensory pattern determination processing unit **224**. That is, the chromatic image feature amounts corresponding to each of the sensory patterns acquired in step **S2205** are compared with the image feature amounts extracted in step **S2203** to calculate their matching level. In this case, the matching levels for all sensory patterns stored in the image feature amount/sensory pattern correspondence holding unit **223** are calculated. The matching level is calculated using a scheme such as vector computations, statistic processes, or the like using cosine measure.

In step **S2207**, the matching levels between all the sensory patterns and the image calculated in step **S2206** are stored in the sensory pattern holding unit **220** in correspondence with the image ID of that image. This example is as has already been described previously with reference to FIG. **5**.

The aforementioned process is done for all images to be registered.

As described above, according to this embodiment, since both the feature amount of multimedia information itself

corresponding to a query word which indicates multimedia information wanted, and the content word that describes the contents of multimedia information are used as query conditions on the basis of associative words associated with the query word, desired multimedia information wanted can be accurately extracted.

For example, in a conventional system, when "sea" is obtained as a word which is associated with a query word "refreshing", a search result "rough sea" is highly likely to be found. However, in this embodiment, such result is excluded when it is integrated with search results using sensory patterns obtained from a combination "refreshing"—"sea".

Since multimedia information can be searched based on associative words that express the contents pertaining to a query word indicating desired multimedia information, and the feature amount of multimedia information itself is used, multimedia information having an inappropriate feature amount which cannot meet the query word can be accurately extracted.

In the above embodiment, image information is used as information wanted. As for multimedia information (e.g., audio information) other than image information, the present invention can be applied by executing information feature amount extraction, and corresponding the extracted information feature amount to sensory patterns.

In the above description, the image holding unit **218**, image content word holding unit **219**, and sensory pattern holding unit **220** which undergo a search are allocated on the DISK **14** that builds a single device, but these building components may be distributed on different devices, and processes may be done on the network via the NIC **19**.

Note that the present invention may be applied to either a system constituted by a plurality of devices (e.g., a host computer, an interface device, a reader, a printer, and the like), or an apparatus consisting of a single equipment (e.g., a copying machine, a facsimile apparatus, or the like).

The objects of the present invention are also achieved by supplying a storage medium, which records a program code of a software program that can implement the functions of the above-mentioned embodiments to the system or apparatus, and reading out and executing the program code stored in the storage medium by a computer (or a CPU or MPU) of the system or apparatus.

In this case, the program code itself read out from the storage medium implements the functions of the above-mentioned embodiments, and the storage medium which stores the program code constitutes the present invention.

As the storage medium for supplying the program code, for example, a floppy disk, hard disk, optical disk, magneto-optical disk, CD-ROM, CD-R, magnetic tape, nonvolatile memory card, ROM, and the like may be used.

The functions of the above-mentioned embodiments may be implemented not only by executing the readout program code by the computer but also by some or all of actual processing operations executed by an OS (operating system) running on the computer on the basis of an instruction of the program code.

Furthermore, the functions of the above-mentioned embodiments may be implemented by some or all of actual processing operations executed by a CPU or the like arranged in a function extension board or a function extension unit, which is inserted in or connected to the computer, after the program code read out from the storage medium is written in a memory of the extension board or unit.

As many apparently widely different embodiments of the present invention can be made without departing from the spirit and scope thereof, it is to be understood that the invention is not limited to the specific embodiments thereof except as defined in the appended claims.

What is claimed is:

1. An information search apparatus for managing a plurality of kinds of multimedia information, and searching the managed multimedia information for desired multimedia information, comprising:

a content word holding unit for storing the multimedia information, and content words which verbalize concepts expressed in the multimedia information in correspondence with each other;

an associative word dictionary for storing the content words and associative words which are associated with the content words in correspondence with each other;

input means for inputting a query word;

first search means for acquiring an associative word corresponding to the query word input by said input means from said associative word dictionary, and searching multimedia information on the basis of the acquired associative word;

extraction means for extracting a feature amount corresponding to the query word input by said input means;

second search means for searching multimedia information on the basis of the feature amount extracted by said extraction means; and

integration means for integrating search results obtained by said first and second search means.

2. The apparatus according to claim **1**, wherein said input means can also input a search perspective.

3. The apparatus according to claim **2**, further comprising:

a concept discrimination dictionary for storing index words and antithetic concepts corresponding to the index words in correspondence with each other, and

wherein said first search means acquires an index word and antithetic concept corresponding to the query word from said concept discrimination dictionary on the basis of the query word and search perspective input by said input means, and acquires an associative word corresponding to the query word from said associative word dictionary on the basis of the acquired index word and antithetic concept.

4. The apparatus according to claim **1**, further comprising:

a holding unit for storing associative words and sensory patterns in correspondence with each other, and

wherein said extraction means acquires a sensory pattern corresponding to the associative word, which corresponds to the query word, from said holding unit, and extracts a feature amount of the acquired sensory pattern as the feature amount corresponding to the query word.

5. The apparatus according to claim **1**, wherein the multimedia information is image information.

6. The apparatus according to claim **5**, wherein the feature amount includes at least one of color scheme information, composition information, and shape information contained in the image information.

7. The apparatus according to claim **1**, wherein said integration means integrates the search results obtained by said first and second search means using first matching levels obtained from the search results of said first search means, and second matching levels obtained from the search results of said second search means.

8. An information search method for managing a plurality of kinds of multimedia information, and searching the

managed multimedia information for desired multimedia information, comprising:

the storage step of storing on a storage medium a content word holding unit for storing the multimedia information, and content words which verbalize concepts expressed in the multimedia information in correspondence with each other, and an associative word dictionary for storing the content words and associative words which are associated with the content words in correspondence with each other;

the input step of inputting a query word;

the first search step of acquiring an associative word corresponding to the query word input in the input step from said associative word dictionary, and searching multimedia information on the basis of the acquired associative word;

the extraction step of extracting a feature amount corresponding to the query word input in the input step;

the second search step of searching multimedia information on the basis of the feature amount extracted in the extraction step; and

the integration step of integrating search results obtained in the first and second search steps.

9. The method according to claim **8**, wherein the input step includes the step of allowing to also input a search perspective.

10. The method according to claim **9**, wherein the storage step also includes the step of storing on said storage medium a concept discrimination dictionary for storing index words and antithetic concepts corresponding to the index words in correspondence with each other, and

the first search step includes the step of acquiring an index word and antithetic concept corresponding to the query word from said concept discrimination dictionary on the basis of the query word and search perspective input in the input step, and acquiring an associative word corresponding to the query word from said associative word dictionary on the basis of the acquired index word and antithetic concept.

11. The method according to claim **8**, wherein the storage step also includes the step of storing on said storage medium a holding unit for storing associative words and sensory patterns in correspondence with each other, and

the extraction step includes the step of acquiring a sensory pattern corresponding to the associative word, which corresponds to the query word, from said holding unit, and extracting a feature amount of the acquired sensory pattern as the feature amount corresponding to the query word.

12. The method according to claim **8**, wherein the multimedia information is image information.

13. The method according to claim **12**, wherein the feature amount includes at least one of color scheme information, composition information, and shape information contained in the image information.

14. The method according to claim **8**, wherein the integration step includes the step of integrating the search results obtained in the first and second search steps using first matching levels obtained from the search results in the first search step, and second matching levels obtained from the search results in the second search step.

15. A computer readable memory for storing a program code of an information search process for managing a plurality of kinds of multimedia information, and searching the managed multimedia information for desired multimedia information, comprising:

a program code of the storage step of storing on a storage medium a content word holding unit for storing the multimedia information, and content words which verbalize concepts expressed in the multimedia information in correspondence with each other, and an associative word dictionary for storing the content words and associative words which are associated with the content words in correspondence with each other;

a program code of the input step of inputting a query word;

a program code of the first search step of acquiring an associative word corresponding to the query word input in the input step from said associative word dictionary, and searching multimedia information on the basis of the acquired associative word;

a program code of the extraction step of extracting a feature amount corresponding to the query word input in the input step;

a program code of the second search step of searching multimedia information on the basis of the feature amount extracted in the extraction step; and

a program code of the integration step of integrating search results obtained in the first and second search steps.

16. An information search apparatus for managing a plurality of kinds of multimedia information, and searching the managed multimedia information for desired multimedia information, comprising:

a content word holding unit for storing the multimedia information, and content words which verbalize concepts expressed in the multimedia information in correspondence with each other;

an associative word dictionary for storing the content words and associative words which are associated with the content words in correspondence with each other;

input means for inputting a query word;

a concept discrimination dictionary for storing index words corresponding to the query word and search perspectives pertaining to the index words in correspondence with each other;

display means for extracting search perspectives pertaining to an index word corresponding to the query word input by said input means from said concept discrimination dictionary, and displaying the extracted search perspectives;

designation means for designating a desired one of the search perspectives displayed by said display means;

first search means for acquiring an associative word corresponding to the query word input by said input means from said associative word dictionary, and searching multimedia information on the basis of the acquired associative word;

second search means for extracting a feature amount corresponding to the query word input by said input means, and searching multimedia information on the basis of the extracted feature amount; and

integration means for integrating search results obtained by said first and second search means on the basis of the search perspective designated by said designation means.

17. The apparatus according to claim **16**, wherein said concept discrimination dictionary also stores antithetic concepts corresponding to the index words, and

wherein said first search means acquires an index word and antithetic concept corresponding to the query word

from said concept discrimination dictionary on the basis of the query word and search perspective input by said input means, and acquires an associative word corresponding to the query word from said associative word dictionary on the basis of the acquired index word and antithetic concept.

18. The apparatus according to claim 17, further comprising:

a holding unit for storing associative words and sensory patterns in correspondence with each other, and

wherein said extraction means acquires a sensory pattern corresponding to the associative word, which corresponds to the query word, from said holding unit, and extracts a feature amount of the acquired sensory pattern as the feature amount corresponding to the query word.

19. The apparatus according to claim 17, wherein the multimedia information is image information.

20. The apparatus according to claim 19, wherein the feature amount includes at least one of color scheme information, composition information, and shape information contained in the image information.

21. An information search method for managing a plurality of kinds of multimedia information, and searching the managed multimedia information for desired multimedia information, comprising:

the input step of inputting a query word;

the storage step of storing on a storage medium a content word holding unit for storing the multimedia information, and content words which verbalize concepts expressed in the multimedia information in correspondence with each other, an associative word dictionary for storing the content words and associative words which are associated with the content words in correspondence with each other, and a concept discrimination dictionary for storing index words corresponding to the query word and search perspectives pertaining to the index words in correspondence with each other;

the display step of extracting search perspectives pertaining to an index word corresponding to the query word input in the input step from said concept discrimination dictionary, and displaying the extracted search perspectives;

the designation step of designating a desired one of the search perspectives displayed in the display step;

the first search step of acquiring an associative word corresponding to the query word input in the input step from said associative word dictionary, and searching multimedia information on the basis of the acquired associative word;

the second search step of extracting a feature amount corresponding to the query word input in the input step, and searching multimedia information on the basis of the extracted feature amount; and

the integration step of integrating search results obtained in the first and second search steps on the basis of the search perspective designated in the designation step.

22. The method according to claim 21, wherein said concept discrimination dictionary also stores antithetic concepts corresponding to the index words, and

the first search step includes the step of acquiring an index word and antithetic concept corresponding to the query word from said concept discrimination dictionary on the basis of the query word and search perspective input in the input step, and acquiring an associative word corresponding to the query word from said associative word dictionary on the basis of the acquired index word and antithetic concept.

23. The method according to claim 21, wherein the storage step also includes the step of storing on said storage medium a holding unit for storing associative words and sensory patterns in correspondence with each other, and

the second search step includes the step of acquiring a sensory pattern corresponding to the associative word, which corresponds to the query word, from said holding unit, and extracting a feature amount of the acquired sensory pattern as the feature amount corresponding to the query word.

24. The method according to claim 21, wherein the multimedia information is image information.

25. The method according to claim 24, wherein the feature amount includes at least one of color scheme information, composition information, and shape information contained in the image information.

26. A computer readable memory for storing a program code of an information search process for managing a plurality of kinds of multimedia information, and searching the managed multimedia information for desired multimedia information, comprising:

a program code of the input step of inputting a query word;

a program code of the storage step of storing on a storage medium a content word holding unit for storing the multimedia information, and content words which verbalize concepts expressed in the multimedia information in correspondence with each other, an associative word dictionary for storing the content words and associative words which are associated with the content words in correspondence with each other, and a concept discrimination dictionary for storing index words corresponding to the query word and search perspectives pertaining to the index words in correspondence with each other;

a program code of the display step of extracting search perspectives pertaining to an index word corresponding to the query word input in the input step from said concept discrimination dictionary, and displaying the extracted search perspectives;

a program code of the designation step of designating a desired one of the search perspectives displayed in the display step;

a program code of the first search step of acquiring an associative word corresponding to the query word input in the input step from said associative word dictionary, and searching multimedia information on the basis of the acquired associative word;

a program code of the second search step of extracting a feature amount corresponding to the query word input in the input step, and searching multimedia information on the basis of the extracted feature amount; and

a program code of the integration step of integrating search results obtained in the first and second search steps on the basis of the search perspective designated in the designation step.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 6,493,705 B1
DATED : December 10, 2002
INVENTOR(S) : Kobayashi et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Title page,
Item [56], **References Cited**, OTHER PUBLICATIONS,
"An-Automatic" should read -- An Automatic --; and
"Forth" should read -- Fourth --.

Column 7,
Line 43, "magnetooptical" should read -- magneto-optical --.

Column 16,
Line 29, "52013)," should read -- S2013), --.

Column 17,
Line 49, "value a" should read -- value α --.

Signed and Sealed this

Eleventh Day of November, 2003

JAMES E. ROGAN
Director of the United States Patent and Trademark Office

09/998225

B

(12) **United States Patent**
Cusson et al.

(10) **Patent No.: US 6,487,641 B1**
(45) **Date of Patent: Nov. 26, 2002**

(54) **DYNAMIC CACHES WITH MISS TABLES**

(75) Inventors: **Michael J. Cusson**, Chelmsford, MA (US); **Marcos G. Almeida**, Nashua, NH (US); **Ramu V. Sunkara**, Los Altos, CA (US); **Anil J. D'Silva**, Nashua, NH (US)

(73) Assignee: **Oracle Corporation**, Parkwood Shores, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/654,653**

(22) Filed: **Sep. 5, 2000**

Related U.S. Application Data

(63) Continuation-in-part of application No. 09/294,656, filed on Apr. 19, 1999.
(60) Provisional application No. 60/168,589, filed on Dec. 2, 1999.

(51) **Int. Cl.**[7] **G06F 12/00**
(52) **U.S. Cl.** **711/144**; 711/141; 711/118; 709/220
(58) **Field of Search** 711/141–144, 118; 707/2, 10, 202; 709/202–203, 220; 712/218

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,924,096 A * 7/1999 Draper et al. 707/10
5,940,594 A * 8/1999 Ali et al. 709/203
6,098,064 A * 8/2000 Pirolli et al. 707/2
6,128,627 A * 10/2000 Mattis et al. 707/202
6,148,394 A * 11/2000 Tung et al. 712/218
6,173,311 B1 * 1/2001 Hassett et al. 709/202

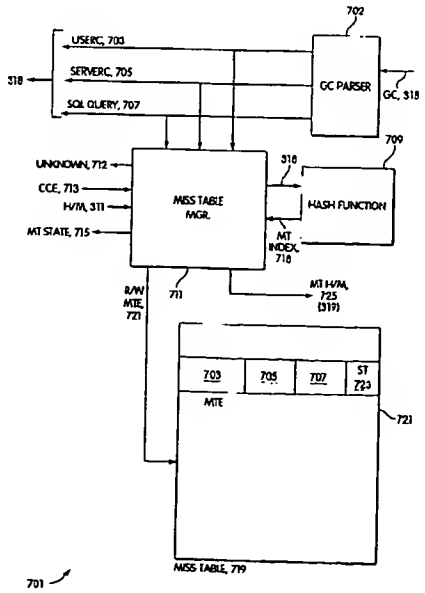
* cited by examiner

Primary Examiner—Do Hyun Yoo
Assistant Examiner—Nasser Moazzami
(74) *Attorney, Agent, or Firm*—Gordon E. Nelson

(57) **ABSTRACT**

A middle-tier Web server with a queryable cache that contains items from one or more data sources. Items are included in the cache on the basis of the probability of future hits on the items. When the data source determines that an item that has been included in the cache has changed, it sends an update message to the server, which updates the item if it is still included in the cache. In a preferred embodiment, the data source is a database system and triggers in the database system are used to generate update messages. In a preferred embodiment, the data access layer determines whether a data item required by an application program is in the cache. If it is, the data access layer obtains the item from the cache; otherwise, it obtains the item from the data source. The queryable cache includes a miss table that accelerates the determination of whether a data item is in the cache. The miss table is made up of miss table entries that relate the status of a data item to the query used to access the data item. There are three statuses: miss, indicating that the item is not in the cache, hit, indicating that it is, and unknown, indicating that it is not known whether the item is in the cache. When an item is referenced, the query used to access it is presented to the table. If the entry for the query has the status miss, the data access layer obtains the item from the data source instead of attempting to obtain it from the cache. If the entry has the status unknown, the data access layer attempts to obtain it from the cache and the miss table entry for the item is updated in accordance with the result. When a copy of an item is added to the cache, miss table entries with the status miss are set to indicate unknown.

16 Claims, 8 Drawing Sheets



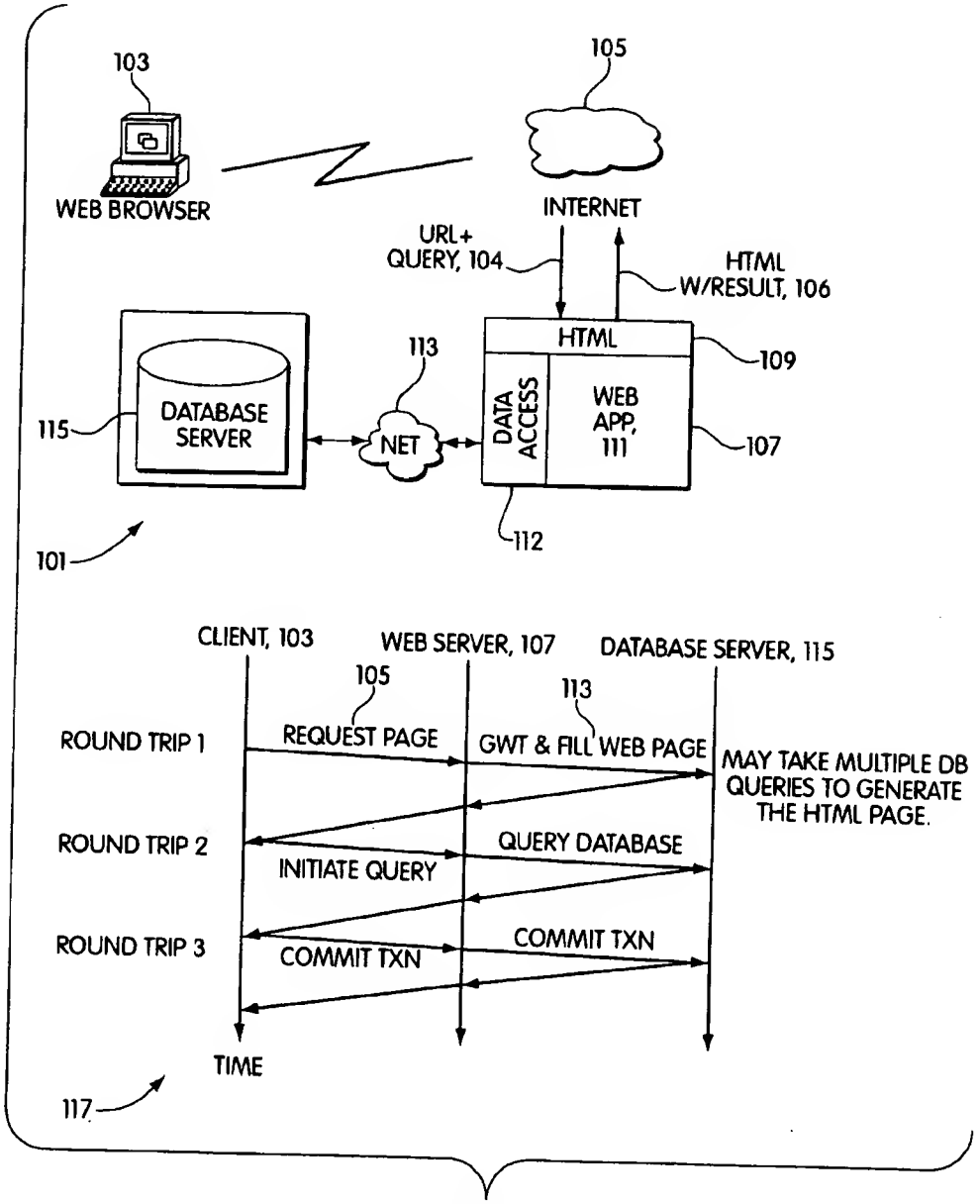


Fig. 1
PRIOR ART

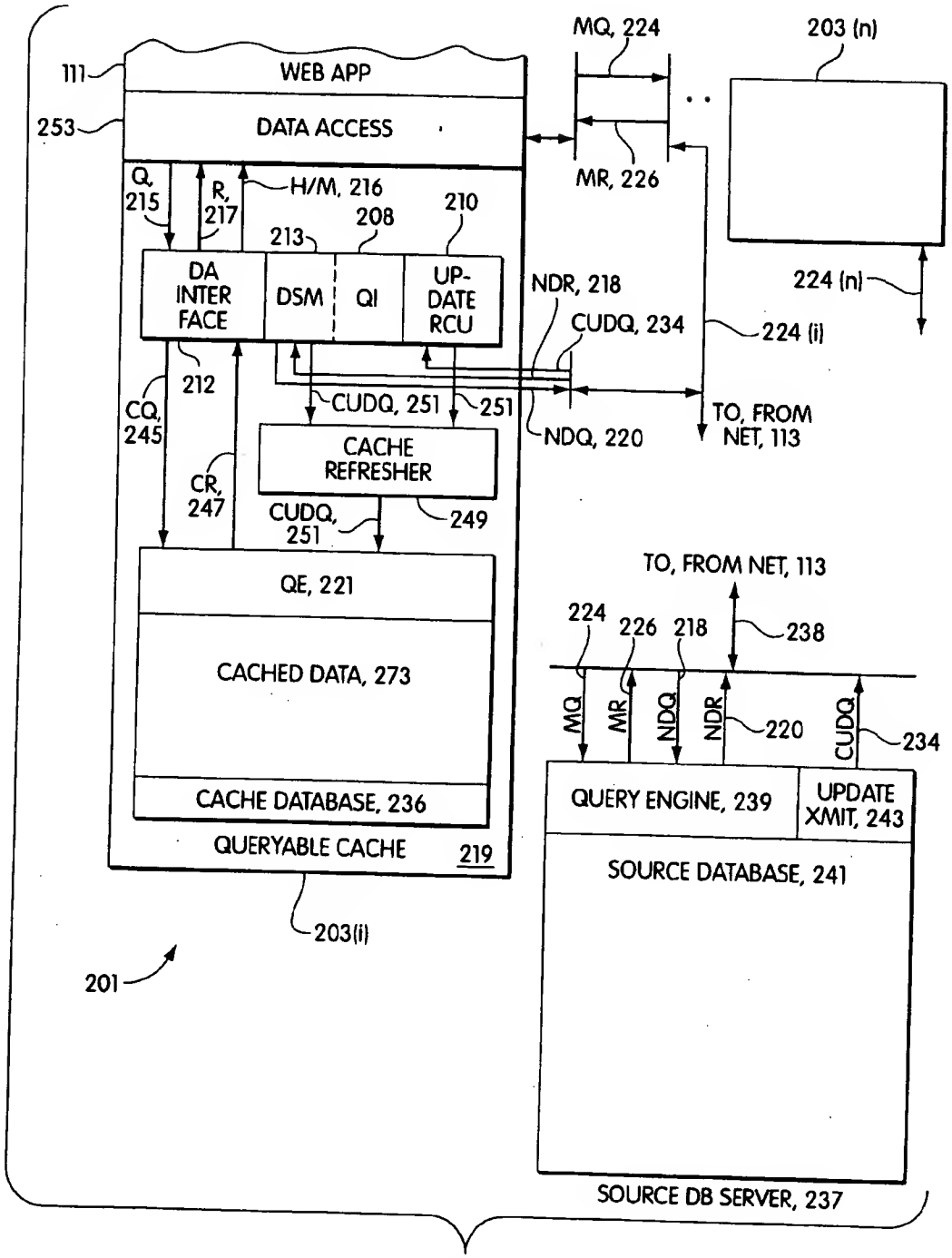

111
WEB APP
253
DATA ACCESS
Q, 215
R, 217
H/M, 216
213
208
210
DA INTER FACE
DSM
QI
UP-DATE RCU
212
CQ, 245
CR, 247
CUDQ, 251
251
CACHE REFRESHER
249
QE, 221
CACHED DATA, 273
CACHE DATABASE, 236
QUERYABLE CACHE 219
203(i)
201
MQ, 224
MR, 226
NDR, 218
CUDQ, 234
NDQ, 220
203 (n)
224 (n)
224 (i)
TO, FROM NET, 113
TO, FROM NET, 113
224
226
218
238
MQ
MR
NDQ
NDR
220
CUDQ
234
QUERY ENGINE, 239
UPDATE XMIT, 243
SOURCE DATABASE, 241
SOURCE DB SERVER, 237

Fig. 2

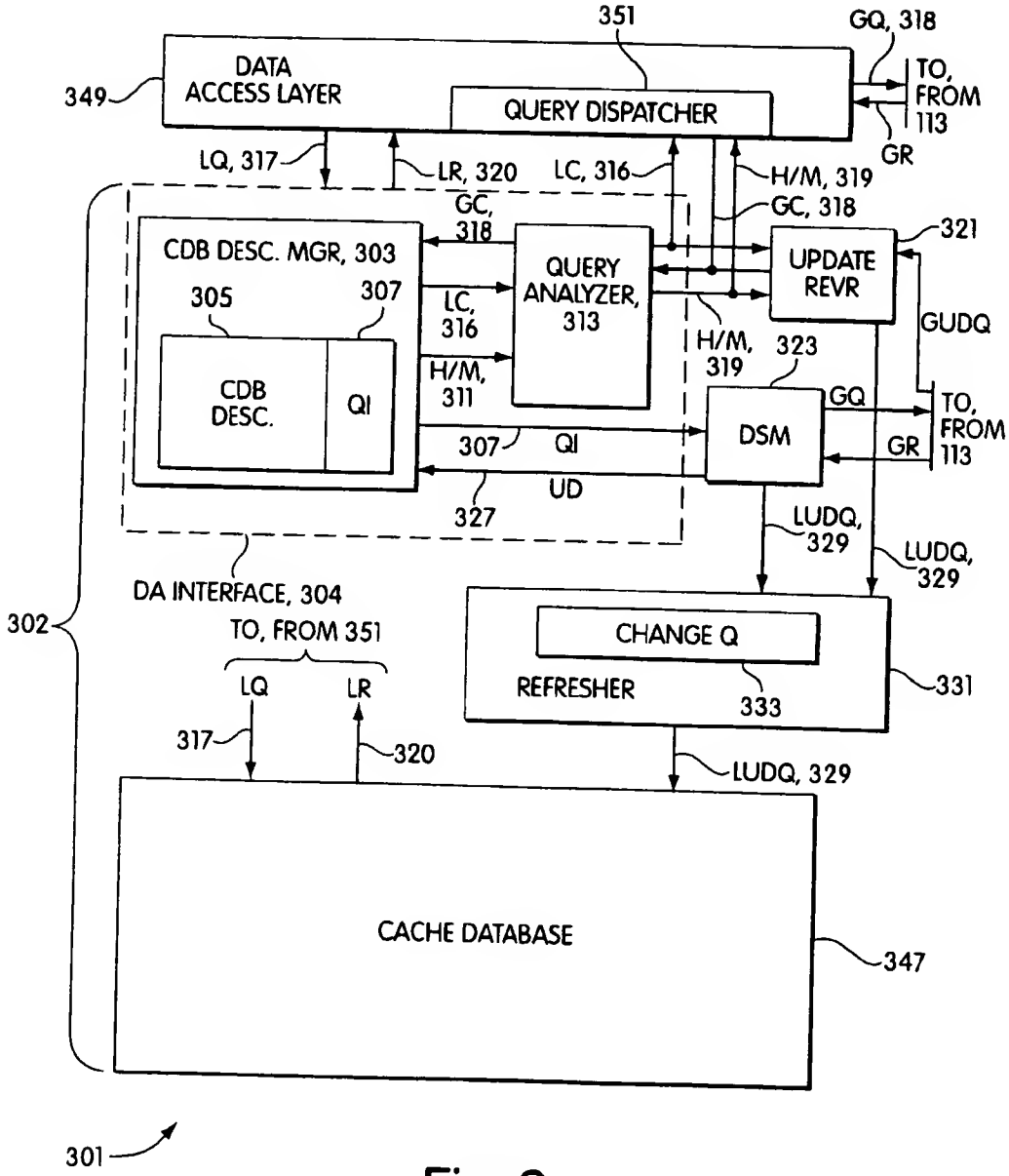

351
GQ, 318
DATA ACCESS LAYER
349
QUERY DISPATCHER
TO, FROM 113
GR
LQ, 317
LR, 320
LC, 316
H/M, 319
GC, 318
GC, 318
321
CDB DESC. MGR, 303
305
307
CDB DESC.
QI
LC, 316
H/M, 311
QUERY ANALYZER, 313
UPDATE REVR
GUDQ
H/M, 319
323
GQ
TO, FROM 113
DSM
GR
307
QI
UD
327
LUDQ, 329
LUDQ, 329
DA INTERFACE, 304
302
TO, FROM 351
LQ
LR
CHANGE Q
REFRESHER
333
331
317
320
LUDQ, 329
CACHE DATABASE
347
301

Fig. 3

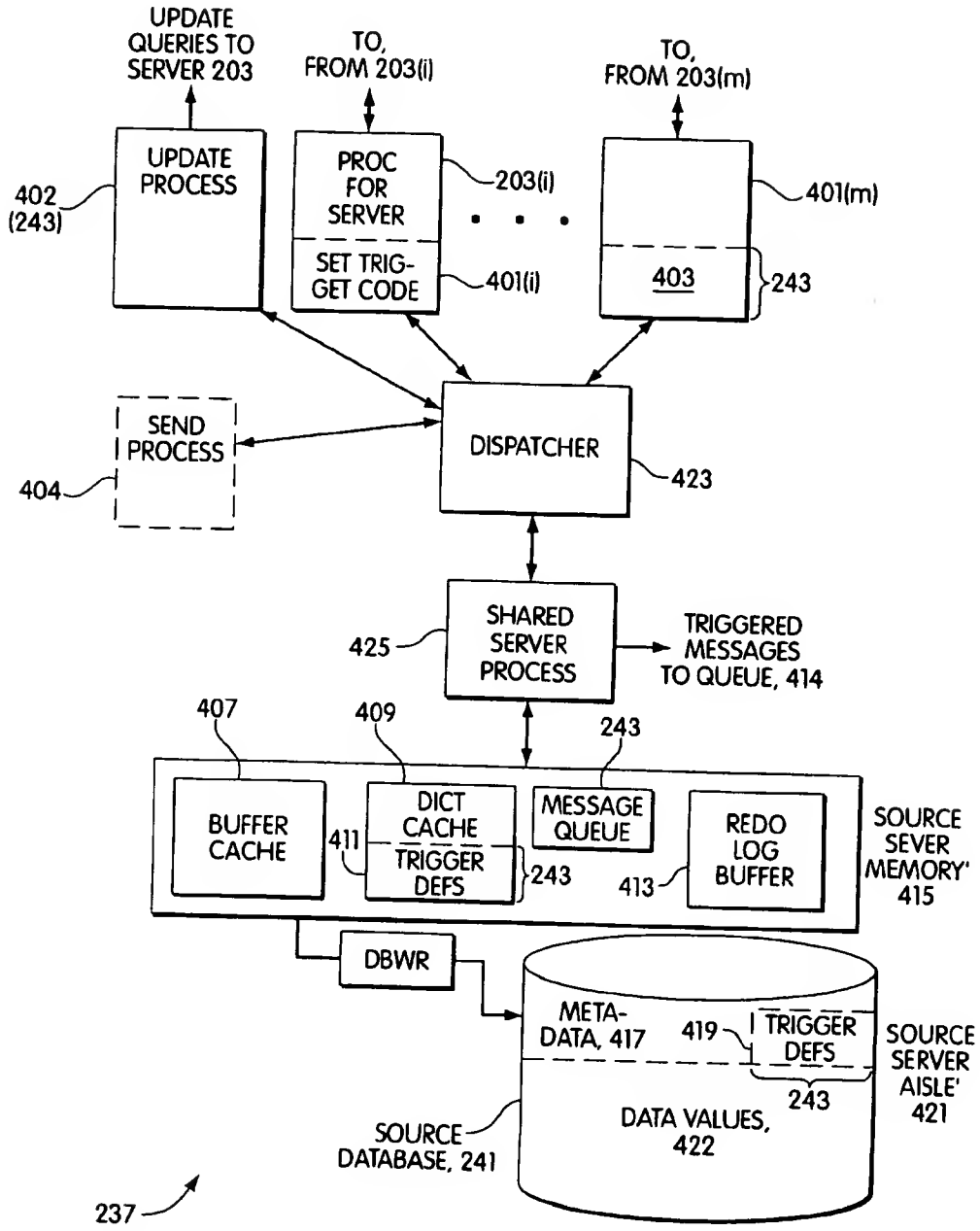

UPDATE QUERIES TO SERVER 203
TO, FROM 203(i)
TO, FROM 203(m)
UPDATE PROCESS
402 (243)
PROC FOR SERVER
203(i)
SET TRIG-GET CODE
401(i)
401(m)
403
243
SEND PROCESS
404
DISPATCHER
423
SHARED SERVER PROCESS
425
TRIGGERED MESSAGES TO QUEUE, 414
407
409
243
BUFFER CACHE
DICT CACHE
411
TRIGGER DEFS
243
MESSAGE QUEUE
413
REDO LOG BUFFER
SOURCE SEVER MEMORY' 415
DBWR
META-DATA, 417
419
TRIGGER DEFS
243
DATA VALUES, 422
SOURCE SERVER AISLE' 421
SOURCE DATABASE, 241
237

Fig. 4

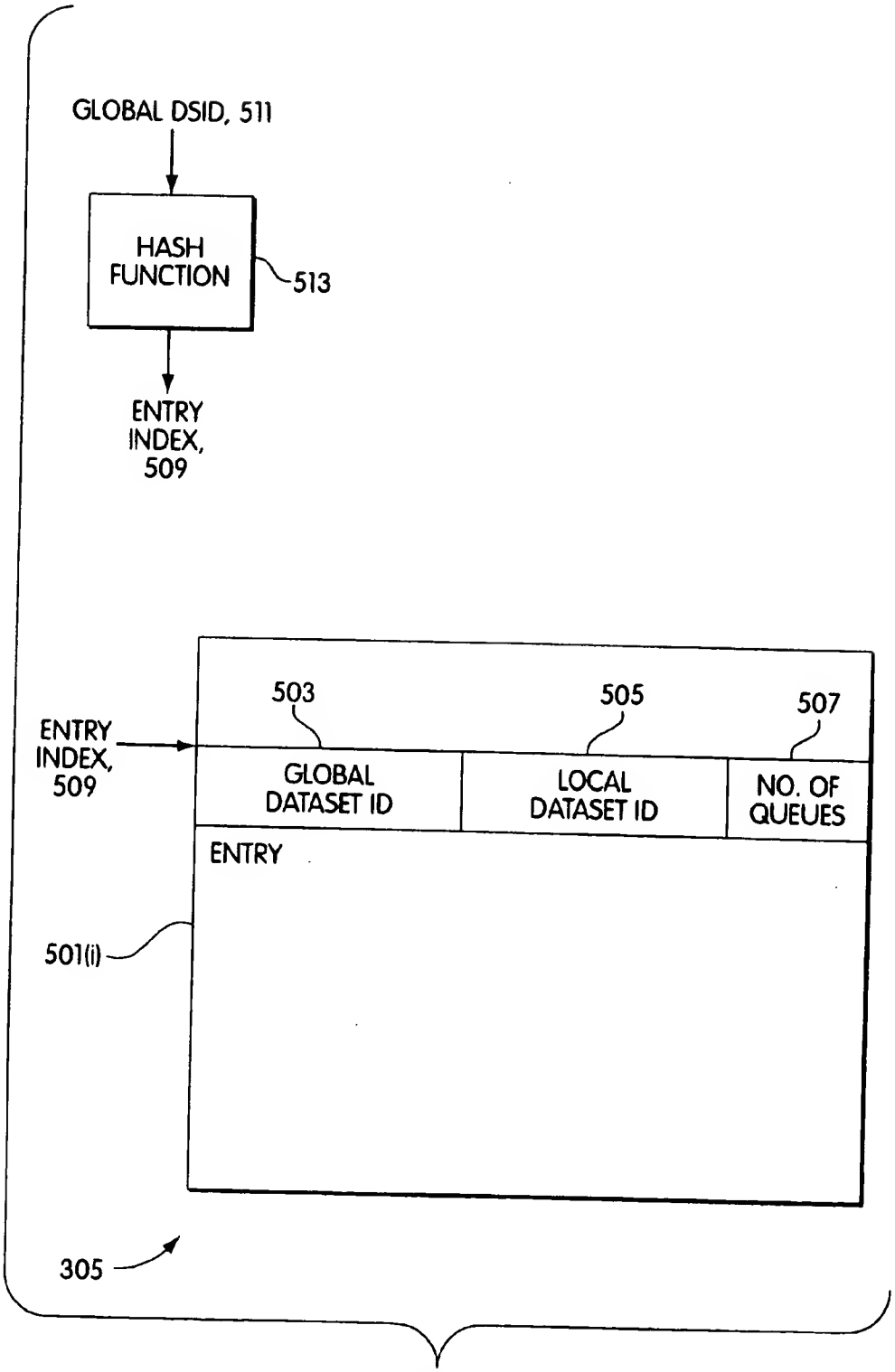

GLOBAL DSID, 511
HASH FUNCTION
513
ENTRY INDEX, 509
503
505
507
ENTRY INDEX, 509
GLOBAL DATASET ID
LOCAL DATASET ID
NO. OF QUEUES
ENTRY
501(i)
305

Fig. 5

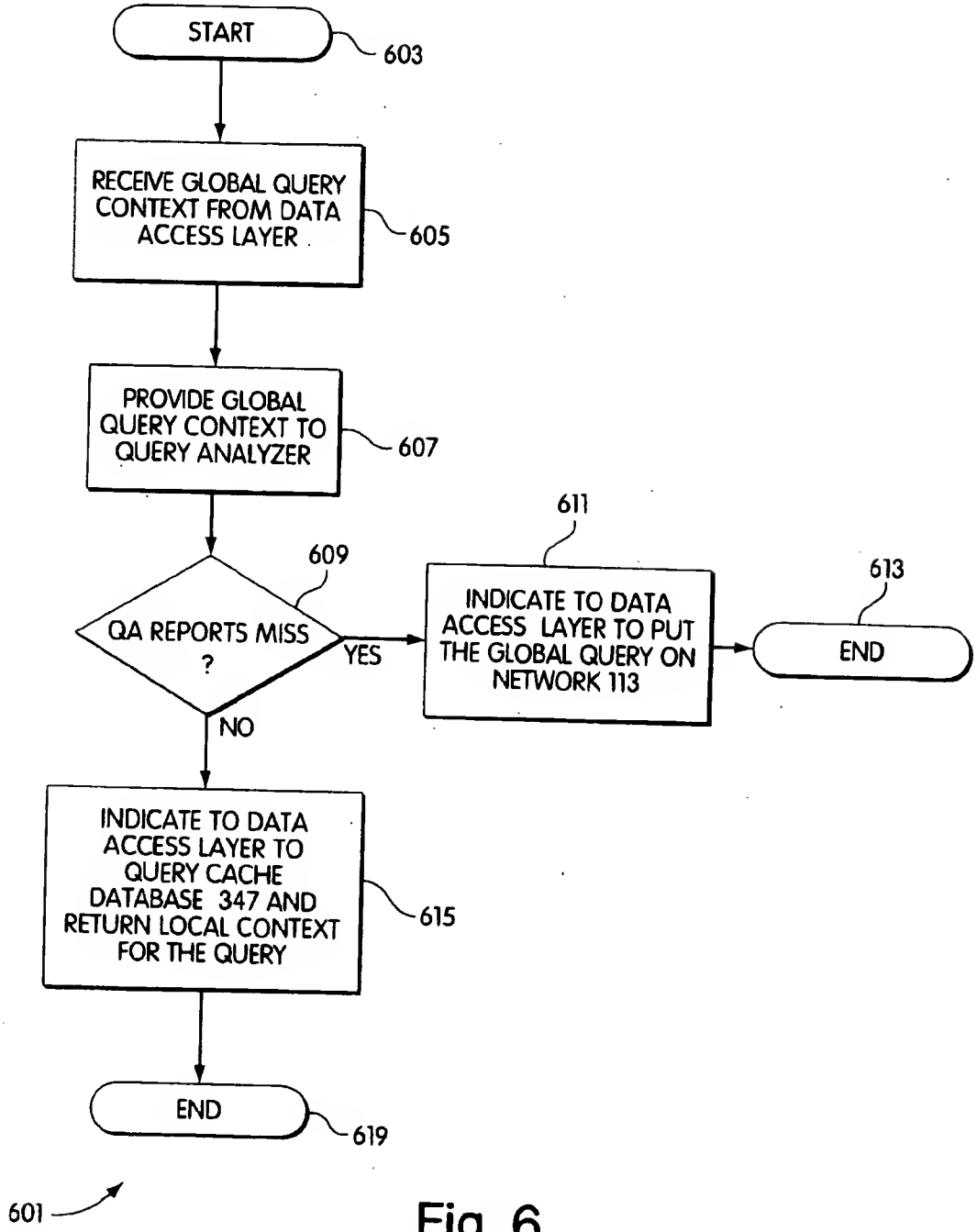

START
603
RECEIVE GLOBAL QUERY CONTEXT FROM DATA ACCESS LAYER
605
PROVIDE GLOBAL QUERY CONTEXT TO QUERY ANALYZER
607
609
QA REPORTS MISS ?
YES
NO
611
INDICATE TO DATA ACCESS LAYER TO PUT THE GLOBAL QUERY ON NETWORK 113
613
END
INDICATE TO DATA ACCESS LAYER TO QUERY CACHE DATABASE 347 AND RETURN LOCAL CONTEXT FOR THE QUERY
615
END
619
601

Fig. 6

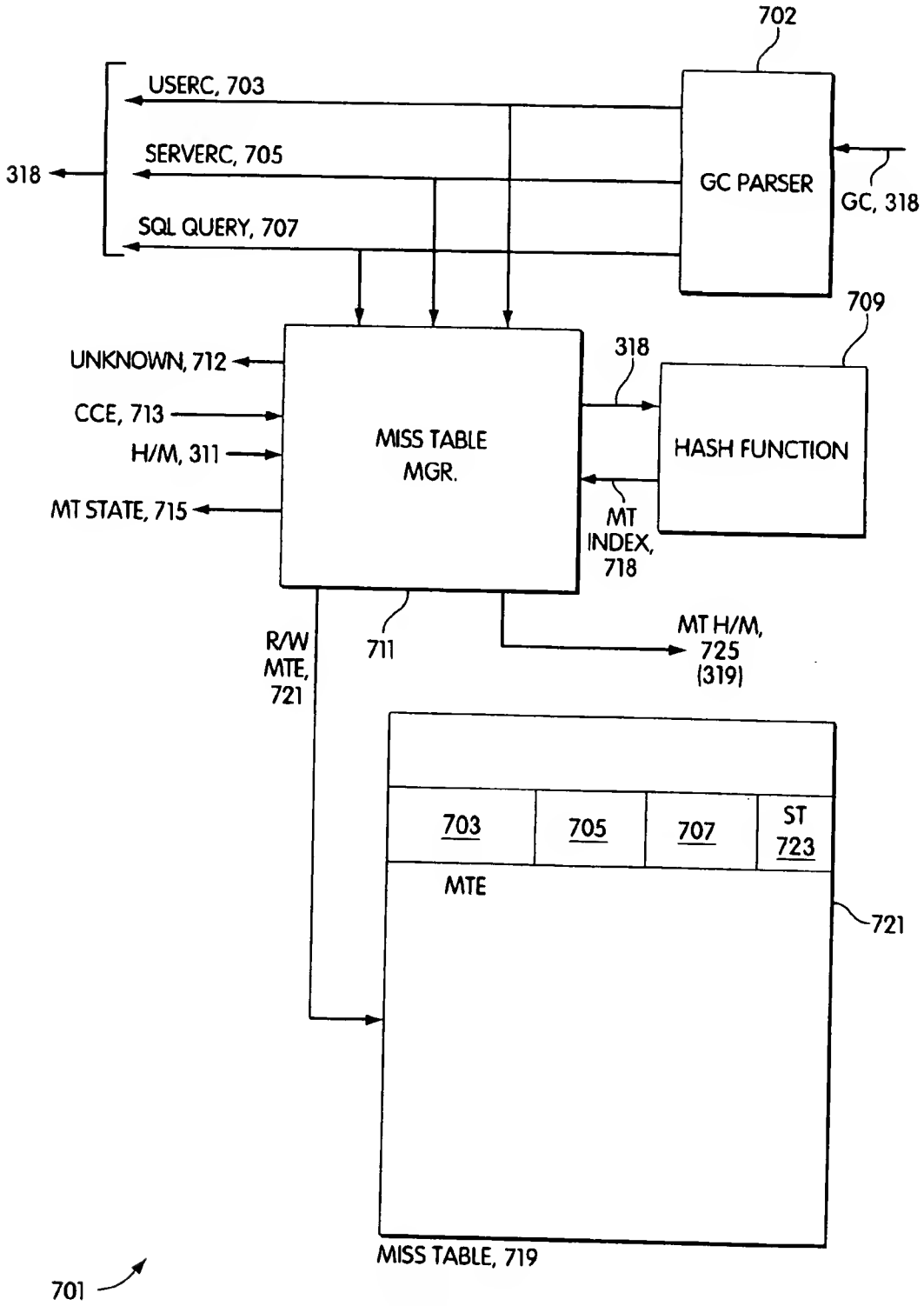

702
USERC, 703
SERVERC, 705
SQL QUERY, 707
318
GC PARSER
GC, 318
709
UNKNOWN, 712
CCE, 713
H/M, 311
MT STATE, 715
MISS TABLE MGR.
318
HASH FUNCTION
MT INDEX, 718
R/W MTE, 721
711
MT H/M, 725 (319)
703
705
707
ST 723
MTE
721
MISS TABLE, 719
701

Fig. 7

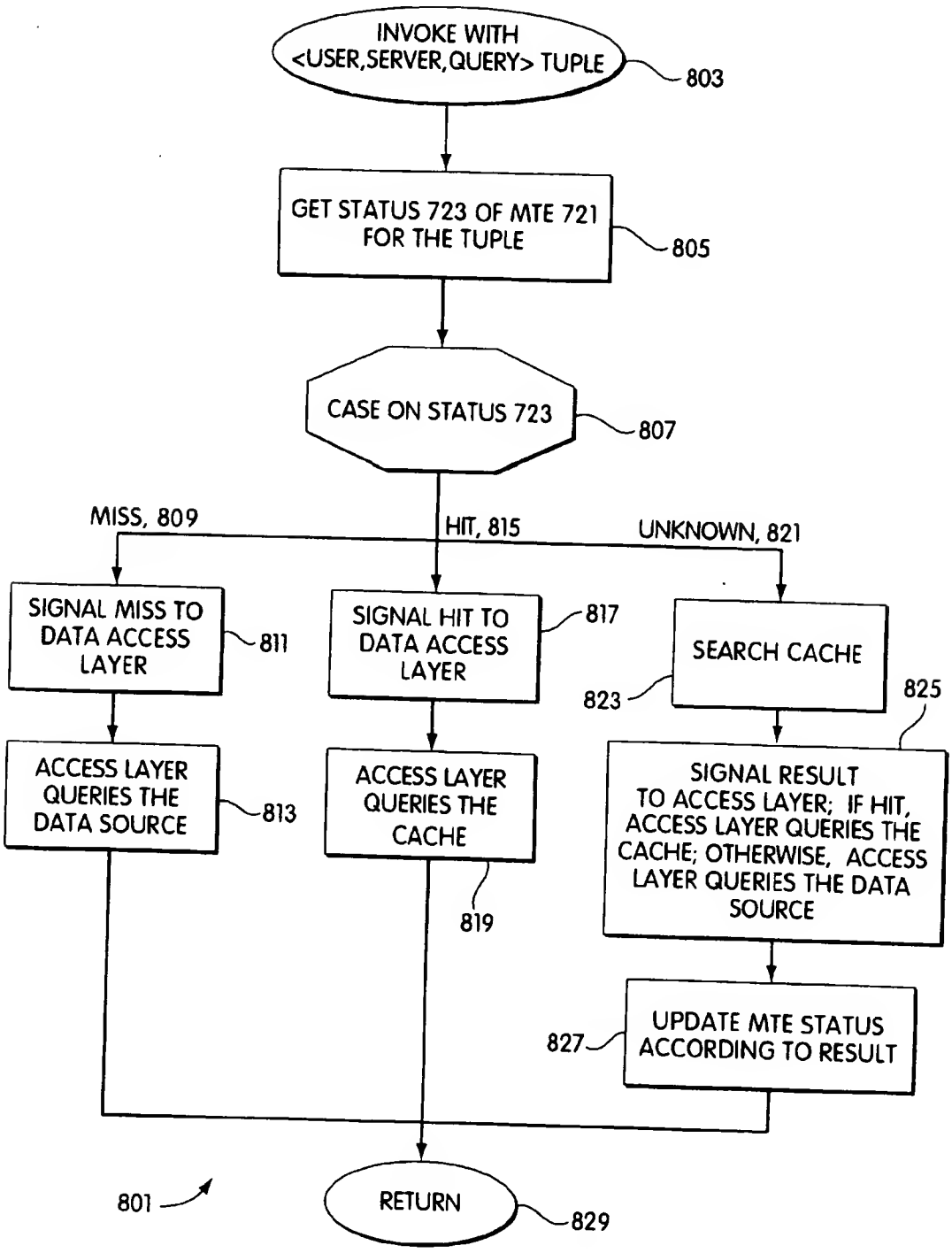

INVOKE WITH <USER,SERVER,QUERY> TUPLE
803
GET STATUS 723 OF MTE 721 FOR THE TUPLE
805
CASE ON STATUS 723
807
MISS, 809
HIT, 815
UNKNOWN, 821
SIGNAL MISS TO DATA ACCESS LAYER
811
SIGNAL HIT TO DATA ACCESS LAYER
817
SEARCH CACHE
823
825
ACCESS LAYER QUERIES THE DATA SOURCE
813
ACCESS LAYER QUERIES THE CACHE
819
SIGNAL RESULT TO ACCESS LAYER; IF HIT, ACCESS LAYER QUERIES THE CACHE; OTHERWISE, ACCESS LAYER QUERIES THE DATA SOURCE
UPDATE MTE STATUS ACCORDING TO RESULT
827
801
RETURN
829

Fig. 8

DYNAMIC CACHES WITH MISS TABLES

CROSS-REFERENCES TO RELATED APPLICATIONS

The present patent application is a continuation-in-part of U.S. Ser. No. 09/294,656, Cusson, et al., Web servers with queryable dynamic caches, filed Apr. 19, 1999, and. claims priority from U.S. Provisional Application No. 60/168,589, Cusson et al., Improving the performance of dynamic data caches by collecting multi-user query miss statistics, filed Dec. 2, 1999. The patent application contains the entire Detailed Description and drawing of U.S. Ser. No. 09/294,656. The new material begins with FIG.7 and the section of the Detailed Description entitled Making cache misses faster.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The invention concerns caching of data in networks generally and more specifically concerns the caching of queryable data in network servers.

2. Description of the Prior Art

Once computers were coupled to communications networks, remote access to data became far cheaper and easier than ever before. Remote access remained the domain of specialists, however, since the available user interfaces for remote access were hard to learn and hard to use. The advent of World Wide Web protocols on the Internet have finally made remote access to data available to everyone. A high school student sitting at home can now obtain information about Karlsruhe, Germany from that city's Web site and a lawyer sitting in his or her office can use a computer manufacturer's Web site to determine what features his or her new PC ought to have and then configure, order, and pay for the PC.

A consequence of the new ease of remote access and the new possibilities it offers for information services and commerce has been an enormous increase in the amount of remote access. This has in turn lead to enormous new burdens on the services that provide remote access and the resulting performance problems are part of the reason why the World Wide Web has become the World Wide Wait.

FIG. 1 shows one of the causes. of the performance problems. At **101**, there is shown the components of the system which make it possible for a user at his or her PC to access an information source via the World Wide Web. Web browser **103** is a PC which is running Web browser software. The Web browser software outputs a universal resource locator (URL) **104** which specifies the location of a page of information in HTML format in the World Wide Web and displays HTML pages to the user. The URL may have associated with it a message containing data to be processed at the site of the URL as part of the process of obtaining the HTML page. For example, if the information is contained in a database, the message may specify a query on the data base. The results of the query would then be returned as part of the HTML page. Internet **105** routes the URL **104** and its associated message to the location specified by the URL, namely Web server **107**. There, HTML program **109** in Web server **107** makes the HTML page **106** specified by the URL and returns it to Web browser **103**. If the message specifies a query on the database in database server **115**, HTML program **109** hands the message off to Web application program **111**, which translates the message into a query in the form required by data access layer **112**.

Data access layer **112** is generally provided by the manufacturer of database server **115**. It takes queries written in standard forms such as OLE-DB, ODBC, or JOBC, converts the queries into the form required by database server **115**, and places the queries in messages in the form required by network **113**. Database server **115** then executes the query and returns the result via network **113** to data access layer **112**, which puts the results into the required standard form and returns them to Web application **111**, which in turn puts the result into the proper format for HTML program **109**. HTML program **109** then uses the result in making the HTML page **106** to be returned to browser **103**.

As may be seen from the above description, a response to a URL specifying a page whose construction involves database server **115** requires four network hops: one on Internet **105** from browser **103** to Web server **107**, one on network **113** from server **107** to server **115**, one on network **113** from server **115** to server **107**, and one on Internet **105** from server **107** to browser **103**. If more than one query is required for an HTML page, there will be a round trip on network **113** for each query.

Moreover, as shown at **117**, a typical Web transaction is a series of such responses: the first HTML page includes the URL for a next HTML page, and so forth. The transaction shown at **117** begins with a request for an HTML page that is a form which the user will fill out to make the query; data base server **115** provides the information for the HTML page. When that page is returned, the user fills out the form and when he or she is finished, the browser returns a URL with the query from the form to server **107**, which then deals with the query as described above and returns the result in another HTML page. That page permits the user to order, and when the user orders, the result is another query to database server **115**, this time, one which updates the records involved in the transaction.

Not only do Web transactions made as shown in FIG. 1 involve many network hops, they also place a tremendous burden on data base server **115**. For example, if data base server **115** belongs to a merchant who sells goods on the Web and the merchant is having a special, many of the ;transactions will require exactly the same sequence of HTML pages and will execute exactly the same queries, but because system **101** deals with each request from a web browser individually, each query must be individually executed by database server **115**.

The problems of system **101** are not new to the designers of computer systems. There are many situations in a computer system where a component of the system needs faster access to data from a given source, and when these situations occur, the performance of the system can be improved if copies of data that is frequently used by the component are kept at a location in the system to which the component has faster access than it has to the source of the data. When such copies exist, the location at which the copies are kept is termed a cache and the data is said to be cached in the system.

Caching is used at many levels in system **101**. For example, browser **103** keeps a cache of previously-displayed HTML pages, so that, it can provide a previously-displayed HTML page to the user without making a request for the page across Internet **105**. Web server **107** similarly may keep a cache of frequently-requested HTML pages, so that it can simply return the page to the user, instead of constructing it. Database server **115**, finally, may keep a cache of the information needed to answer frequently-made queries, so that it can return a result more quickly than if it were starting

from scratch. In system **101**, the most effective use of caching is in Web server **107**, since data that is cached there is still accessible to all users of internet **105**, while the overhead of the hops on data access **113** is avoided.

Any system which includes caches must deal with two problems: maintaining consistency between the data in the cache and the data in the data source and choosing which data to cache. In system **101**, the first problem is solved in the simplest way possible: it is the responsibility of the component using the data to determine when it needs a new copy of the data from the data source. Thus, in browser **103**, the user will see a cached copy of a previously-viewed HTML page unless the user specifically clicks on his browser's "reload" button. Similarly, it is up to HTML program **109** to determine when it needs to redo the query that provided the results kept in a cached HTML page. The second problem is also simply solved: when a new page is viewed or provided, it replaces the least recently-used cached page.

Database systems such as the Oracle8™ server, manufactured by Oracle Corporation and described in Leverenz, et al., Oracle8 Server Concepts, release 8.0, Oracle Corporation, Redwood City, Calif., 1998., move a copy of a database closer to its users by replicating the original database at a location closer to the user. The replicated data base may replicate the entire original or only a part of it. Partial replications of a database are termed table snapshots. Such table snapshots are read-only. The user of the partial replication determines what part of the original database is in the table snapshot. Consistency with the original database is maintained by snapshot refreshes that are made at times that are determined by the user of the table snapshot. In a snapshot refresh, the table snapshot is updated to reflect a more recent state of the portion of the original database contained in the snapshot. For details, see pages 30–5 through 30–11 of the Leverenz reference.

There are many applications for which the solution of letting the component that is doing the caching decide when it needs a new page causes problems. For example, when the information in a data source is important or is changing rapidly (for example, stock prices), good service to the user requires that the information in the caches closely tracks the information in the data source. Similarly, there are many situations where caching all data that has been requested causes problems. For instance, in a cache run according to least recently-used principles, any HTML page that is produced by HTML program **109** or received in browser **103** is cached and once cached, stays in the cache and takes up space that could be used for other HTML pages until it attains least recently-used status.

When Web server **107** includes a Web application **111** involving a database server **115**, there is still another problem with caching in web server **107**: since the data is cached in the form of HTML pages, it is not in queryable form, that is, a cached HTML page may contain data from which another query received from Web browser **103** could be answered, but because the data is contained in an HTML page instead of a database table, it is not in a form to which a query can be applied. Thus, even though the data is in server **107**, server **107** must make the query, with the accompanying burden on data base server **115** and delays across network **113**, and the HTML page containing the result of the query must be separately cached in server **107**.

U.S. Ser. No. 09/294,656, Cusson, et al., Web servers with queryable dynamic caches, describes a web server **107** that has a cache in which cached data is to the extent possible in queryable form, in which the cached data is dependably updated when the data in the source changes, and in which selection of data from a source for caching is based on something other than the mere fact that a URL received from a web browser referenced the data, and thus provides a solution to the foregoing problems. The cache thus solves many of the problems of prior-art caches in network environments.

A remaining problem, however, is that the only way that Web server **107** can determine whether a query can be performed on the cache instead of on database server **115** is by doing the query on the cache and if a miss results, doing the query on database server **115**. A query that goes to database server **115** as a result of a cache miss is thus substantially slower than one that goes directly to database server **115**, and when there is a substantial number of cache misses, the result may be a substantial degradation of the overall performance of Web server **107** with a cache. It is an object of the present invention to make a query to database server **115** that results from a cache miss substantially as fast as a query that goes directly to database server **115**.

SUMMARY OF THE INVENTION

The object is achieved by adding a miss table to a cache that contains copies of remotely-stored items. The query that is applied to the cache is in effect a specifier for the item that will be returned by the query. There may or may not be a copy of the item in the cache. If there is not, the remotely-stored item must be fetched. The miss table relates the specifier for the item to a status indicator that indicates at least whether the item is present in the cache. A dispatcher receives the specifier for the item and presents it to the miss table; if the miss table indicates that there is no copy of the item in the cache, the dispatcher uses the item specifier to fetch the remotely-stored data item.

The status indicator may further indicate that it is unknown whether there is a copy of the item in the cache. When the status indicator so indicates, the cache responds to the remote item specifier and provides an indication whether there is a copy of the item in the cache. A miss table manager for the miss table responds to the indication by updating the miss table in accordance with the indication. The cache further provides the miss table manager with a change event notification to the miss table manager when the contents of the cache have changed and the miss table manager responds thereto by setting the status for at least those items for which the status in the miss table indicates that there is no copy and which are affected by the change to unknown.

In an preferred embodiment, the miss table is employed in a network server that includes a cache. The cache may contain a copy of a rowset from a remote location and responds to a rowset specifier specifying the remote location by returning the rowset when there is a copy in the cache. The miss table relates the rowset specifier to a status indicator as described above. If the miss table indicates to the network server that there is no copy of the rowset in the cache, the network server fetches the rowset from the remote location.

Other objects and advantages will be apparent to those skilled in the arts to which the invention pertains upon perusal of the following Detailed Description and drawing, wherein:

BRIEF DESCRIPTION OF THE DRAWING

FIG. **1** is an example of a prior-art system for performing queries via the World Wide Web;

FIG. 2 is a high-level block diagram of a system of the invention;

FIG. 3 is a detailed block diagram of details of an implementation of server 203;

FIG. 4 is a detailed block diagram of details of an implementation of source database server 237;

FIG. 5 is a detail of cache database description 305;

FIG. 6 is a flowchart of the operation of query dispatcher 351;

FIG. 7 is an overview of a miss table as used in query analyzer 313 of queryable cache 302; and

FIG. 8 is a flowchart of the operation of the miss table.

Reference numbers in the drawing have three or more digits: the two right-hand digits are reference numbers in the drawing indicated by the remaining digits. Thus, an item with the reference number 203 first appears as item 203 in FIG. 2.

DETAILED DESCRIPTION

The following Detailed Description will begin with a conceptual overview of the invention and will then describe a presently-preferred embodiment of the invention.

Overview of the Invention: FIG. 2

FIG. 2 shows a system 201 for retrieving information via a network which includes one or more network servers 203(0 . . . *n*). Each server 203(*i*) includes a queryable cache 219 that is automatically updated when information cached in cache 223 changes in source database 241 and in which the contents of cache 223 are determined by an analysis of what queries will most probably be made by users of server 203(*i*) in the immediate future. Server 203 is a Web server 107, and thus has an HTML component 109, a Web application component 111, and a data access component 253 which is a version of data access component 112 which has been modified to work with queryable cache 219. Server 203 could, however, communicate with its users by any other kind of network protocol. Server 203 further communicates with source data base server 237 by means of network 213, which may use any protocol which is suited to the purpose.

FIG. 2 shows one server 203, server 203(*i*), in detail. As before, Web application 111 provides a query in a standard form to data access 253. Here, however, data access 253 has access not only to source database server 237 via network 113, but also to queryable cache 219, which contains a cache data base 236 that has a copy 223 of a portion of the data in source database 241. When data access 253 receives a query from web application 111, it first presents the query to queryable cache 219, as shown at Q 215. If cached data 223 includes the data specified in the query, queryable cache 219 returns result (R) 217, which data access 253 returns to Web application 111. If cached data 223 does not include the data specified in the query, queryable cache 219 returns a miss signal (M) 216 to data access 253, which then makes the query via network 113 to source database server 237 and when it receives the result, returns it to Web application 111. The query made in response to the miss signal appears as miss query (MQ) 224 and the response appears as miss response (MR) 226.

It is important to note here that because the interactions with queryable cache 219 and with source database server 237 are both performed by data access layer 253, the existence of queryable cache 219 is completely transparent to Web application 111. That is, a Web application program 111 that runs on Web server 107 will run without changes on Web server 203(*i*).

Continuing in more detail with queryable cache 219, the data cached in queryable cache 219 is contained in cache database 236, which, like any database, contains data, in this case, copies of datasets (database tables) from source database 241 that are cached in queryable cache 219,.and a query engine (QE 221), which runs queries on the datasets in cached data 223. The portion of queryable cache 219 which receives queries from data access layer 253 is data access interface 212. Data access interface 212 has two functions:

- It determines whether the query can be executed on cached data 223 required to execute query 215 and generates miss signal 216 if it does not.
- If cached data 223 does contain the data, it puts query 215 into the proper form for cache database 236.

Data access interface 212 makes the determination whether the query can be executed by analyzing the query to determine the query's context, that is, what datasets are required to execute the query and then consulting a description of cached data 223 to determine whether these datasets are present in cached data 223. The datasets are specified in the query by means of dataset identifiers, and consequently, the context is for practical purposes a list of the identifiers for the required data sets. The description 223 of course includes the dataset identifiers for the cached data sets. If the required datasets are present, data access interface 212 makes cache query 245, which has the form required to access the data in cache data base 236. Cache database 236 returns cache result 247, which data access interface 212 puts into the form required for result 217.

Because cached data 223 is contained in cache database 236, cached data 223 is queryable, that is, if a dataset is contained in cached data 223, queryable cache 219 can return as a result not only the entire dataset, but any subset of that dataset that can be described by a query. For example, if cached data 223 includes a dataset that lists all of the kinds of shirts sold by a company engaged in Web commerce and the list of kinds includes the colors that each kind of shirt is available in, queryable cache 219 will be able to handle a query for which the result is a list of the kinds of shirt that are available in red.

Cached data 223 is kept consistent with source database 241 by means of update transmitter 243 in source database server 237 and update receiver 210 in queryable cache 219. Whenever a change occurs in source database 241 in a dataset of which there may be a copy in cached data 223, update transmitter 243 generates a cache update query (CUDQ) 234 specifying the change and sends CUDQ 234 via network 113 to each of servers 203(0 . . . *n*). Update receiver 210 receives CUDQ 234 from network 113 and determines from the data set description maintained by DA 212 whether the dataset is in face in cached data 223; if it is, it puts the cache update query into the proper form 251 for cache database 236 and provides it to cache refresher 249, which then runs update query 251 on cache database 236.

Data set manager (DSM) 213 decides generally what copies of datasets from source database server 237 are to be included in cache database 236. The information that DSM 213 uses to make this determination is contained in query information 208. Query information 208 may be any information available to server 203(*i*) which can be used to predict what datasets of source database 241 will most probably be queried in the near future. For example, if a company engaged in Web commerce is having a 1-day sale on certain items for which there are datasets in source database 241, query information 208 may indicate the datasets for the items and the time of the 1-day sale. Using that information, DSM 213 can obtain the datasets from

source database **241** and cache them in cache database **236** before the beginning of the sale and remove them from cache database **236** after the end of the sale.

Another kind of query information **208** is a query log, a time-stamped log of the queries received from data access layer **253**; if the log shows a sharp increase in the occurrence of queries for a given dataset, DSM **213** should cache the datasets for that query in cache **219** if they are not there already. Conversely, if the log shows a sharp decrease in the occurrence of such queries, DSM **213** should consider removing these datasets from queryable cache **219**. When DSM **213** determines that a dataset should be added to queryable cache **219**, it sends a new data query (NDQ) **218** via network **113** to source data base **241** to obtain the new data and when DSM **213** has the response (NDR **220**), it sends a delete query to query engine **221** indicating the data to be deleted in cached data **223** to make way for the new data and then sends a cache update query **251** to cache refresher **249** to update the cache.

Data set manager **213** and query information **208** may also be implemented in part in source data base server **237** or anywhere where information about the probability of future queries may be obtained. When implemented in source data base server **237**, the query log would log each query **231** to source database **241** and at least the portion of data set manager **213** which reads the query log to determine what new data needs to be cached would be in source database server **237**; when it determined that new data needed to be cached, it would send an update query with the new data to each of the servers **203**. The component of DSM **213** that determines what is to be removed could also be in source database server **237**, in which case, all queryable caches **219** would contain the same data in cached data **223**, or that component could be in each server **203**(*i*), with the component making decisions concerning what data to remove to accommodate the new data based on the present situation in server **203**(*i*). In such an arrangement, there can be a local query log in each server **203** in addition to the global query log in source database server **241**. Such an arrangement would permit different servers **203** to have different-sized caches **223**; it would also permit different servers **203** to take local variations in the queries they are receiving into account in determining what data to remove from cache **219**. One way such variations might occur is if system **201** were set up so that different servers **203** preferentially received queries from users in different geographical locations.

FIG. **2** shows only a single source database server **237**; there may of course be more than one; moreover, source database server **237** need not be a classical database system. Server **203**(*i*) can be set up to be used with data sources containing any kind of queryable data, where queryable is defined as having a form which can be represented as a set of numbered rows of data. Such a set of numbered rows is termed a rowset. Database tables are of course one example of rowsets; others are files of data records, text files, and still and moving image data. If server **203**(*i*) is used with data sources having only a single kind of queryable data, queryable cache **219** need only be set up to deal with that kind of queryable data.

If server **203**(*i*) is used with data sources having more than one kind of queryable data, cache database **236** may be set up using a rowset representation that will accommodate all of the different kinds of queryable data. In that case, DA **212**, DSM **213**, and update receiver **210** will translate between the results and update queries received from the various data sources and the representations used in cached data **236**. In other embodiments, there may be more than one cache database **236** in queryable cache **219**, with different cache databases being used for different kinds of queryable data. Again, DA **212**, DSM **213**, and update receiver **210** will perform the necessary translations.

Details of a Preferred Embodiment of a Data Access Layer and a Queryable Cache FIGS. 3, 5, and 6

FIG. **3** shows a preferred embodiment **301** of data access **349** and queryable cache **302**. Corresponding components of FIGS. **2** and **3** have the same names. Cache database **347** in embodiment **301** is an Oracle8 Server, which is described in detail in Leverenz, et al., Oracle8 Server Concepts, release 8.0, Oracle Corporation, Redwood City, Calif., 1998. In preferred embodiment **301**, Web application **111** uses global data set identifiers in queries. The Web applications **111** in all of the servers **203** use the same set of global data set identifiers. A cache data base **347** in a given server **203** has its own set of local data set identifiers for the data sets cached in cache data base **347**. In preferred embodiment **301**, then, one may speak of global queries and query contexts that use global data set identifiers and local queries and query contexts that use local data set identifiers. In the preferred embodiment, query analyzer **313** uses cached data base description **305** to translate global query contexts into local query contexts.

Data access layer **349** includes a new component, query dispatcher **351**, which is the interface between data access layer **349** and queryable cache **302**. FIG. **6** is a flowchart **601** of the operation of query dispatcher **351** in a preferred embodiment. Reference numbers in parentheses refer to elements of the flowchart. When data access layer **349** is preparing to query source database **241**, it provides the global context for the query to query dispatcher **351** (**605**), which in turn provides global context **318** (FIG. **3**) to query analyzer **313** (**607**). Query analyzer **313** determines whether the datasets identified by the global context are cached in cache database **347**; if they are not, query analyzer **313** reports a miss **319** to query dispatcher **351** (**609**), which indicates to data access layer **349** that it is to place the global query on network **113**.

If the datasets identified by the global context are cached in cache database **347**, query analyzer **313** indicates that fact to query dispatcher **351** and also provides query dispatcher **351** with local context **316** for the datasets in cache database **347** (**615**). Query dispatcher **351** then provides the local context to data access layer **349**, which uses the local context to make a local query **317** corresponding to the global query and then uses the local query to obtain local result **320** from cache database **347**. It should be noted here that the operations involved in the translation from the global query to the local query and applying the local query to cache database **347** may be divided among data access layer **349**, query dispatcher **351**, and query analyzer **313** in many different ways; the advantage of the technique of flowchart **601** is that data access layer **349** can employ the same mechanisms to make local queries as it does to make global queries. All query analyzer **313** and query dispatcher **351** need do is supply data access layer **349** with the local context needed to make the local query.

Continuing with the details of queryable cache **302** and beginning with DA interface **304**, interface **304** receives a global context **318** from query dispatcher **351** and depending on whether the datasets for the queries are in cache database **347**, provides either local context **316** or a miss signal **319**.

DA interface **304** has two main components: query analyzer **313** and cache database description manager **303**.

Query analyzer **313** analyzes global contexts received from data access layer **253** and other components of embodiment **301** to obtain the global context's global dataset identifiers. Having obtained the global dataset identifiers, query analyzer **313** provides them to CDB description manager **303**, which looks them up in cache database description **305**. Cache database description **305** is a table of datasets. At a minimum, there is an entry in the table for each dataset that has a copy in cache database **347**. Each such entry contains the dataset's global identifier and its local identifier. The table also contains query information **307**. CDB description manager **303** then returns an indication of whether the dataset is in cache database **347** (H/M **311**). If it is not, the query cannot be run on cache database **347**, but must be run on source database **241**, and consequently, query analyzer **313** returns a miss signal **319** to query dispatcher **351**. If the query can be run on cache database **347**, query analyzer **313** returns a hit signal **319** and also returns local context **316** for the query. As indicated above, query dispatcher **351** then provides local context **316** to data access layer **349**, which uses it to make local query **317** on cache database **347**. Cache database **347** then returns local result **320** to data access layer **349**.

FIG. **5** shows details of CDB description **305**. In a preferred embodiment, it is a table which has at least an entry **501** for each dataset of source database **241** of which here is a copy in cache database **347**. Each entry **501** contains the global dataset identifier for the data set, by which the dataset is known in all servers **107** with queryable caches **219** containing copies of the dataset, the local data set identifier **505**, by which the dataset is known in cache database **347**, and number of queries **507**, which indicates the number of times the dataset has been queried over an interval of time. In the preferred embodiment, number of queries **507** embodies query information **307**.

An entry **501**(*i*) for a given dataset is accessed in a preferred embodiment by a hash function **503**, which takes global dataset ID **507** for the dataset and hashes it into an entry index **509** in table **305**. CDB description manager **303** then searches table **305** for the entry **501** whose field **503** specifies global DSID **511** beginning at entry index **509**. If no such entry is found, the dataset is not in cache database **347** and CDB description manager **303** signals a miss **311** to query analyzer **313**. Table **305** may also include entries **501** for global datasets that are not presently cached in cache database **347**; in such entries, local dataset ID **505** has a null value and a miss is returned in response to the null value. The purpose of such entries is to maintain number of queries information **507** for such data sets, so that dataset manager **323** can determine whether to add the entry's dataset to cache database **347**.

Update Rcvr **321** receives update queries provided by source database server **237** from data access **253** and uses query analyzer **313** to determine whether the dataset affected by the update is in cache database **347**. If it is not, update rcvr **321** ignores the update; otherwise, it places update query **329** in change queue **333**. Refresher **331** reads queue **333** and executes its queries.

Data store manager **323** uses query information **307** in CDB description **305** to determine what datasets to add to or delete from cache database **347**. With datasets to be added, DSM **323** makes the necessary queries to source database **241** and when the results arrive, DSM **323** makes them into update queries **239** and provides tbe update queries **329** to change queue **333**, from which they are executed by refresher **331** as described above. DSM **323** further updates CDB description **305** as required by the changes it makes in cache database **347**, as shown at **327**.

In a preferred embodiment, DSM **323** and refresher **331** have their own threads or processes. It should also be pointed out here that CDB description **305** and change queue **333** could be implemented as database tables in cache database **347**. Because tbese components are implemented independently of cache database **347** and because abstract query translator **339** is used as an interface to cache database **347**, embodiment **301** is to a large extent independent of the particular kind of database system employed to implement cache database **347**. In embodiment **301**, data access **203** only provides read queries to data access interface **304**. All update queries go directly to server **237**, without the update being entered in cache database **347**. In other embodiments, queryable cache **219** may be implemented as a writethrough cache, i.e., the update may be entered in cache database **347** and also sent to server **237**. It should be pointed out here that most Web applications are mostly-read applications, that is, a Web user typically spends far more time reading information than he or she does changing it. For instance, in Web commerce, the "shopping" is mostly a matter of reading HTML pages, with updates happening only wben the user adds something to his or her "shopping cart" or makes his or her purchases. In a system such as system **201**, only making the purchases would typically involve an update of source database **241**.

Details of Source database Server **237**: FIG. **4**

FIG. **4** shows a preferred embodiment of source database server **237**. Source database server **237** in the preferred embodiment is implemented by means of an Oracle8 server executing on a computer system that includes a disk drive **421** on which is stored source database **241** and memory **415** which contains buffer cache **407** for copies of data values **421** from database **241** and dictionary cache **409** for copies of metadata from database **241**. Metadata is database tables whose contents describe the data in the database. Writebacks of cached data in server memory **415** to source database **241** are handled by database write process **325**. Each of processes **401**(**0** . . . *n*) represents and corresponds to a server **203** and handles queries resulting from cache misses, update queries, and queries from DSM **323** in the corresponding server **203**. Dispatcher **311** gives each of these processes in turn access to shared server process **317**, which performs the actual queries and returns the results to the querying process, which in turn returns the results via network **235** to its corresponding server **203**.

The Oracle8 implementation of source database server **237** is a standard Oracle8 database system to which has been added an implementation of update transmitter **243**, which automatically sends an update to queryable cache **219** in each of the servers **203**(**0** . . . *n*) when data in source database **241** that has been copied to cached data **223** changes. The components of updater **243** in FIG. **4** are labeled with the reference number **243** in addition to their own reference numbers. The implementation of updater **243** in the preferred embodiment employs database triggers. A database trigger is a specification of an action to be taken if a predefined change occurs in a data value or an item of metadata in the database. Many database systems permit definition of triggers; triggers in the Oracle8 database system are described in detail at pages 17–1 through 17–17 of the Leverenz reference.

In the preferred embodiment, when a process **401**(*i*) corresponding to a server **203**(*i*) receives a query from DSM

323 in server 203(*i*) for data to be added to server 203(*i*)'s cached data 223, process 203(*i*) executes set trigger code 403. This code sets an Oracle8 AFTER row trigger in metadata 417 for each row of data and/or metadata specified in the query. Shared server process 317 takes the action specified in the trigger whenever the trigger's row of data has been modified. The action specified for the trigger is to send a message to each of the servers 203(0 . . . *n*) with an update query that modifies the data in cached data 223 in the same fashion as it was modified in source database 241. In the preferred embodiment, the action performed by the trigger is to place the message with the update query in message queue 414, which is implemented as an Oracle8 advanced queue. Message queue 414 is read by update process 402, which sends the messages in queue 414 to each of the servers 203(0 . . . *n*).

Adding new data to cached data 223 in response to or in anticipation of changes in the behavior of the users of internet 105 and updating cached data 223 in response to changes in source database 241 may of course be implemented in many other ways in the preferred embodiment shown in FIGS. 3 and 4. For example, determining what data should be in cached data 223 could be done in source DBS server 237 instead of in each of the servers 203. Source database 241, like the cached databases 347 in the servers 203(0 . . . *n*), can maintain statistics information, and a send process 404 in source server 237 can analyze the statistics in substantially the same fashion as described for DSM 323, determine what data should be sent to the servers 203(0 . . . *n*) for caching in cached data 223, make update queries for that data, and place messages containing the update queries in message queue 414, from which update process 402 can send them to the servers 203.

Updating cached data 223 in response to changes in source database 241 can also be implemented without triggers. The Oracle8 database system includes a redo log 413 in source server memory 415 which is a circular buffer of updates that have been performed on source database 241. The database system maintains the log so that it can redo updates in case of system failure, but the log can also be used to update cached data 223. If there is a table in source database 241 which describes cached data 223, update process 402 can use the table in conjunction with redo log 413 to determine whether an update in redo log affects cached data 223. If it does, update process 402 can send a copy of the update query to the servers 203 as just described.

Caching Servers and Source Servers That Do Not Involve Database Systems

The techniques used to determine what data should be cached in server 203 and to update cached data 223 can also be employed in systems where the data is not queryable. For example, the source data may simply be a collection of documents, identified perhaps by a document number (such as its URL, if the document is an HTML page), and the cached data may be simply a subset of the collection. What cache web application 211 would receive from HTML component 109 in such a system would simply be the document number for a document; if it is present in the cached data, the caching server would return it from there; otherwise, it would fetch it from the source server. Query log 205 in such a case would be a time-stamped list of the documents that had been requested, together with an indication of whether the document was in the cached data. DSM 213 in such an embodiment would determine as described above for the database whether a document should be included in the cached data, and having made the determination, would obtain it from the source server. As also described above, a send component on the source server could make the same determination and send the document to the caching servers.

For update purposes, the source server would simply maintain a list of the documents that were presently in the caching servers; if one of the documents on the list was updated, updater 243 would send the new version of the document to the caching servers, where DSM 213 would replace any copy of the document in the cache with the new copy. The techniques just described for documents could of course also be used with files and with audio, image, and motion picture data.

Making Cache Misses Faster

Most dynamic caches are "load on miss" caches, that is, the system to which the cache belongs presumes that data that is being referenced is in the cache, and if a miss occurs, indicating that the data is not in the cache, the system finds the data, fetches it, and loads it into the cache. In a "load on miss" cache, only the first reference to uncached data results in a miss and on the miss, the time it takes to determine whether a miss has occurred is not important compared with the time it takes to find, fetch, and load the data into the cache.

Cache 302, on the other hand, is not a "load on miss" cache. As described above, what data is contained in cache 302 is determined by dataset manager 323, which uses query information 307 along with other information to determine what data will most probably be queried in the near future. In some cases, data set manager 323 may not even put frequently-queried data into cache 302. For example, if the data in question is changing at a rapid rate in source database 241, the cost in system resources of always directing the query to source database 241 may be less than the cost of updating cache 302 every time the data changes in source database 241.

Because a miss on cache 302 does not result in data being loaded into the cache, the cost of determining whether a miss has occurred is not "lost" in the cost of finding, fetching, and loading the data into the cache. Moreover, because a miss does not result in data being loaded, there will generally be more misses than in a "load on miss" cache. For both of these reasons, there is a need in cache 302 to reduce the cost of a miss so that in general, a query to source database 241 that comes about as a result of a miss on cache 302 takes little or no more time than a query that is simply made directly to source database 241.

FIG. 7 shows the apparatus used in a preferred embodiment of cache 302 to reduce the cost of the miss. Cache miss accelerator 701 is a component of query analyzer 313. The chief component of cache miss accelerator 701 is miss table 719, which contains a number of miss table entries 721. Each miss table entry 721 represents a query made to a rowset in a source server 237. The entry 721 for a given query and source server contains a state value 723 which indicates one of at least the following:

- a hit: the rowset specified by the query is in cache 302;
- a miss: the rowset specified by the query is not in cache 302;
- unknown: it is unknown whether the rowset specified by the query is in cache 302.

If a query has a miss table entry 721 in miss table 719 and state value 723 indicates that the rowset specified by the query is not in cache 302, there is no need to search cache database description table 305 to determine whether the

rowset is in the cache. The difference in the amount of time it takes to make a query to source database **241** that results from a miss and the amount of time it takes to make a direct query to source database **241** is simply the time it takes to find the query's entry **721** in miss table **719**.

The remaining components of cache miss accelerator **701** serve to accelerate the process of finding a query's entry **721** in miss table **719** and to maintain miss table **719**. As described above, a query as originally received in data access layer **349** has a global context **318**, which specifies the query in the terms required for source database **214**. A query to cache database **347** must, however, specify the query in the terms required for cache database **347**. These terms are the query's local context **316**. One of the tasks of query analyzer **313** is to parse the global context into its components, so that CDB description manager **303** can translate the global context **318** into the corresponding local context **316**. This parsing task is performed by GC parser **702**, which divides global context **318** into user context **703**, which identifies the user(s) making the query, server context **705**, which identifies source database **214**, and SQL query **707**, which is the SQL statement specifying the query to be made on source database **214**.

All three components of global context **318** are necessary to completely characterize a query, so all three are provided to accelerator **701**. Each miss table entry **721** corresponds to a <user, server, query>tuple. Thus, each miss table entry **721** includes in addition to status field **723** a field **703** indicating the user context, a field **705** indicating a source database **214**, and a field **707** indicating an SQL query. An important property of accelerator **701** is that there is only one miss table entry **721** for a given <user, server, query>tuple. ST **723** in that miss table entry thus makes the experience of any entity making a query with the given tuple available to all entities making queries with the given tuple. This property of accelerator **701** is particularly important in situations where the user is identified by an Internet protocol address that is shared by many different actual users.

Miss table manager **711** is a collection of routines which provides the interface between miss table **711** and the rest of query analyzer **313**. Miss table manager **711**'s most frequent operation is reading the value of status **723** in miss table entry **721** for a given source database and query and reporting whether the entry indicates a hit or miss to query analyzer **313**, which in turn passes the report to data access layer **349**. The operation proceeds as follows: When a user wishes to query the source database, the user provides the global query to data access layer **349**. Query dispatcher **351** then provides global context **318** to query analyzer **313**. Parser **702** parses global context **318** into its components, and all three components go to miss table manager **711**, as shown in FIG. **701**. To ensure rapid access, miss table **719** is implemented as a hash table, using techniques that are well-known in the art. Miss table manager **711** provides UserC **703***k* ServerC **705**, and SQL query **707** to a hash function **709**, which returns an index value **718** corresponding to tuple <**703,705,707**>. If there is a MTE **721** for that tuple <**703,705,707**>, miss table manager **711** can quickly locate the entry using index value **718**. To determine whether the proper MTE **721** has been found, miss table manager **711** compares the values of the tuple <**703,705, 707**> in the MTE **721** with the values of the tuple <**703, 705,707**> received from GC parser **702**.

Once miss table manager **711** has located the proper MTE **721**, miss table manager **711** reads status value **723**. If it indicates that the rowset corresponding to the entry's tuple <**703,705,707**> is not in cache **302**, miss table manager **705** reports a miss to query analyzer **313**, which passes it on to data access layer **349**, as indicated by arrow **725**, which, as indicated by the reference number in parentheses, performs the function of hit/miss indicator **319** of FIG. **3**. Query analyzer **313** further responds to the miss by not passing global context **318** on to CDB description manager **303**

If the proper MTE **721**'s state value **723** indicates a hit, miss table manager **711** reports that as well to query analyzer **313**, which passes the result to data access layer **349**. These components then function as previously described, with query analyzer **313** providing global context **318** to CDB description manager **303**, CDB description manager **303** returning local context **316** to query analyzer **313**, and query analyzer **313** providing it to data access layer **349**, which uses local context **316** to make a local query **317** to query cache database **347**.

If the proper MTE **721**'s state value **723** indicates "unknown", miss table manager **711** reports that fact to query analyzer **313**, as indicated by arrow **712**, and query analyzer **313** provides CDB description manager **303** with MTE **721**'s tuple <**703,705,707**>. CDB description manager **303** then searches CDB descriptions **305**, and CDB description manager **303** provides H/M **311** to query analyzer **313** according to the results of the search. Query analyzer **313** then provides H/M **311** to miss table manager **711**. If H/M **311** indicates that the rowset represented by tuple <**703,705, 707**> is in cache database **347**, miss table manager **711** sets status field **723** in MTE **721** to indicate a hit; query analyzer **313** also indicates a hit via **319** to data access layer **349** and uses CDB description manager **303** to obtain local context **316** for data access layer **349**. If H/M **311** indicates a miss, query analyzer **313** provides the miss to miss table manager **711** and to data access layer **349**, and miss table manager **711** sets status field **723** to indicate a miss.

If there is no entry in miss table **719** for tuple <**703,705, 707**> from global context **318**, miss table manager **711** makes an entry and places it at the index returned by hash function **709** for the tuple. State **723** is set to "unknown" in the entry and miss table manager **711** then proceeds as described above for MTEs **721** with state values **723** indicating "unknown."

In a preferred embodiment, there is a fixed number of MTEs **721** and when a new MTE **721** is required but none is available in miss table **719**, the miss table manager selects a MTE **721** currently in the table for reuse. The selection is done on the basis of least frequent use. Miss table manager **711** keeps track of the frequency of use of MTEs **721** and of the MTE's status **723**. A use of an MTE **721** with a status of "miss" has a much higher weight than a use of an MTE with the status "hit" or the status "unknown", and thus MTEs **721** with the status "miss" tend to stay in miss table **729** longer than MTEs **721** that have one of the other statuses.

In addition to making MTEs **721** and providing and setting the status values in MTEs **721**, miss table manager **711** has to ensure that the contents of miss table **719** track the contents of CDB description **305**. In a preferred embodiment, this is done in response to a cache change event which CDB description manager **303** provides to query analyzer **313** whenever a copy is added to database **347**. Query analyzer **313** provides the cache change event to miss table manager **711**, which responds by setting state value **723** in all of the MTEs in miss table **719** whose status value **723** indicates "miss" to "unknown". As queries corresponding to the MTEs **721** come in, miss table manager **711** deals with those MTEs **721** as described above for MTEs **721** with the status value "unknown". In other embodiments, CDB description manager **303** may provide a

description of the added copy along with the cache change event, and in such embodiments, miss table manager **711** may only change the status of MTEs **721** representing queries that are affected by the presence of the added copy.

In a preferred embodiment, the status value **723** of MTE **721** for a tuple <**703,705,707**> is part of query information **307**; consequently, as indicated by arrow **715**, miss table manager **711** periodically provides the state of all of the entries in MTE **721** to CDB description manager **303**.

The semantics of miss table **719** for the case when there is an entry in miss table **719** corresponding to tuple <**703, 705,707**> from the query are shown in overview in flowchart **801** of FIG. **8**. The relevant routine of miss table manager **711** is invoked at **803** with the <user,server,query>tuple; at **805**, Status **723** for the corresponding MTE **721** is retrieved from miss table **719**; at **807**, the value of status **703** is used to determine whether miss branch **809**, hit branch **815**, or unknown branch **821** will be taken. In miss branch **809**, the miss is signaled to data access layer **349** at **811** and at **813**, the data access layer queries the data source. In hit branch **815**, the hit is signaled to data access layer **349** at **817** and at **819**, the data access layer queries the cache. In unknown branch **821**, the cache is searched for the rowset corresponding to the query at **823**; at **825**, the results of the search are signaled to the data access layer, which queries the cache or the data source accordingly. At **827**, status **723** is updated in the MTE in accordance with the result. The routine returns at **829**.

Conclusion

The foregoing Detailed Description has disclosed to those skilled in the art of caching data how a miss table may be used to speed up references to uncached data in a dynamic cache which is not loaded on miss and how such a miss table may be used with a queryable cache in a network server. The Detailed Description has further disclosed the best mode presently known to the inventors of practicing their invention. It will, however, be immediately apparent to those skilled in the art of caching data that many features of the embodiment of the miss table disclosed in the Detailed Description are consequences of the environment provided by the queryable cache for which it has been implemented and that embodiments can be made for other environments which will work according to the principles of the miss table disclosed herein, but will otherwise differ substantially from the embodiment disclosed herein. It is also the case that many different techniques are known in the art for implementing tables and for accelerating the process of locating an entry in a table, and many of these techniques can be used in implementations of miss tables that work according to the principles of the miss table disclosed herein. Moreover, in other embodiments, the miss table may have statuses in addition to the miss status that differ from those of the miss table disclosed herein and different techniques may be used to decide when a table entry is to be reused or how the miss table should be updated when the data in the cache changes.

For all of the foregoing reasons, the Detailed Description is to be regarded as being in all respects exemplary and not restrictive, and the breadth of the invention disclosed here in is to be determined not from the Detailed Description, but rather from the claims as interpreted with the full breadth permitted by the patent laws.

What is claimed is:

1. An improved network server of a type that includes a cache containing a copy of a rowset from a remote location, the network server responding to a rowset specifier specifying the remote location and the rowset therein by providing the copy from the cache when the copy is therein,

the network server having the improvement comprising:

a miss table that relates the rowset specifier to a status indicator, the status indicator being able to indicate at least whether the copy is in the cache, the network server using the miss table prior to applying the rowset specifier to the cache to determine whether the copy is in the cache, and when not, responding to the rowset specifier by fetching the rowset from the remote location.

2. The network server set forth in claim **1** wherein:

the miss table employs an entry that includes at least the status indicator to relate the rowset specifier to the status indicator; and

the status indicator is further able to indicate that it is unknown whether the copy is in the cache,

the network server responding when the status indicator indicates that it is unknown whether the copy is in the cache by searching for the copy in the cache and setting the status indicator in the entry according to whether the copy is in the cache.

3. The improved network server set forth in claim **2** wherein:

the network server further fetches the copy from the cache when the copy is therein.

4. The improved network server set forth in claim **2** wherein:

when a copy is added to the cache and the status indicator affected thereby currently indicates that the copy is not in the cache, the network server sets at least that status indicator to indicate unknown.

5. The improved network server set forth in claim **4** wherein:

the miss table further comprises a plurality of the miss table entries, each entry having an index; and

when the network server responds to the rowset specifier, the network server hashes the rowset specifier to obtain an index of a miss table entry.

6. The improved network server set forth in any one of claims **1** through **5** wherein:

the cache and the remote location are queryable; and

the rowset specifier specifies a query.

7. An improved method of obtaining a rowset stored in a remote data source in response to a rowset specifier that specifies the remote data source and the rowset therein by performing the steps of

applying the rowset specifier to a local cache to retrieve a copy of the rowset therefrom; and retrieving the rowset from the remote data source only if the copy is not in the local cache, the improved method further comprising the step of:

prior to the step of applying the rowset specifier to the local cache, applying the rowset specifier to a miss table that relates the rowset specifier to a status indicator indicating at least whether the copy is in the local cache,

the step of applying the rowset specifier to the local cache being performed only if the status indicator indicates that the copy is in the local cache.

8. The method set forth in claim **7** wherein

the miss table employs an entry that includes at least the status indicator to relate the rowset specifier to the, status indicator; and

the status indicator is further capable of indicating that it is unknown whether the copy is in the cache; and

the method further comprises the step performed when the status indicator indicates that it is unknown whether the copy is in the cache of:

searching for the copy in the cache and setting the status indicator in the entry according to whether the copy is in the cache.

9. The method set forth in claim 8 further comprising the step of:

fetching the copy from the cache when the copy is therein.

10. The method set forth in claim 8 further comprising the step of:

when a copy is added to the cache and the status indicator affected thereby currently indicates that the copy is not in the cache, setting at least that status indicator to indicate unknown.

11. The method set forth in claim 10 wherein

the miss table further comprises a plurality of the miss table entries, each entry having an index; and

the method further comprises the step of:

hashing the rowset specifier to obtain an index of a miss table entry.

12. The method set forth in any one of claims 7 through 11 wherein:

the cache and the remote location are queryable; and

the rowset specifier specifies a query.

13. Apparatus that fetches items of data from a remote location, the apparatus comprising:

a cache that stores copies of the items and provides an item's copy in response to an item specifier for the item;

a miss table that relates an item specifier to a status indicator that indicates at least whether there is a copy of the item specified by the item specifier in the cache; and

a dispatcher that responds to the item specifier by presenting the item specifier to the miss table prior to applying the item specifier to the cache and on receiving an indication from the miss table that there is no copy in the cache, fetching the item of data from the remote location.

14. The apparatus set forth in claim 13 wherein the apparatus further comprises:

a miss table manager that modifies the miss table;

the miss table employs an entry that includes at least the status indicator to relate the rowset specifier to the status indicator;

the status indicator may further indicate that it is unknown whether there is a copy of the item; and

when the status indicator indicates that it is unknown whether there is a copy of the item, the cache responds to the item specifier and provides an indication whether there is a copy of the item to the miss table manager, the miss table manager updating the miss table in accordance with the indication.

15. The apparatus set forth in claim 14 wherein:

the cache provides a change event notification to the miss table manager when a copy of a data item has been added to the cache; and

when the status indicator affected thereby currently indicates that the copy is not in the cache, the miss table manager responds to the change event notification by setting the status indicator to indicate unknown.

16. The apparatus set forth in any one of claims 13 through 15 wherein:

the cache and the remote location are queryable; and

the item specifier specifies a query.

* * * * *

US006385605B1

(12) **United States Patent**
Suzuoka et al.

(10) **Patent No.: US 6,385,605 B1**
(45) **Date of Patent: May 7, 2002**

(54) **INFORMATION RETRIEVAL APPARATUS AND A METHOD**

(75) Inventors: **Takashi Suzuoka**, Kanagawa-ken; **Nobuyuki Sawashima**, Tokyo, both of (JP)

(73) Assignee: **Kabushiki Kaisha Toshiba**, Kawasaki (JP)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/156,869**

(22) Filed: **Sep. 18, 1998**

(30) **Foreign Application Priority Data**

Sep. 26, 1997 (JP) 9-261176

(51) **Int. Cl.**[7] **G06F 17/30**
(52) **U.S. Cl.** **707/4**; 707/5
(58) **Field of Search** 707/10, 9, 5, 3, 707/4; 395/706

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,640,553	A *	6/1997	Schultz	707/5
5,717,914	A *	2/1998	Husick et al.	707/5
5,737,734	A *	4/1998	Schultz	707/5
5,802,518	A *	9/1998	Karaev et al.	707/9
5,860,009	A *	1/1999	Uchihira et al.	395/706
5,926,807	A *	7/1999	Peltonen et al.	707/3
5,933,832	A *	8/1999	Suzuoka et al.	707/10
6,055,535	A *	4/2000	Suzuoka et al.	707/10
6,067,539	A *	5/2000	Cohen	707/2
6,154,737	A *	11/2000	Inaba et al.	707/3

FOREIGN PATENT DOCUMENTS

JP 4-326163 11/1992

OTHER PUBLICATIONS

Suzuoka et al., "Performance Debugging Based on Scalability Analysis," Frontiers of Massively Parallel Computation, 1995, Proceedings, pp. 406–413, Feb. 6, 1995.*

* cited by examiner

Primary Examiner—Thomas Black
Assistant Examiner—Charles L. Rones
(74) *Attorney, Agent, or Firm*—Oblon, Spivak, McClelland, Maier & Neustadt, P.C.

(57) **ABSTRACT**

An information retrieval apparatus retrieves information from a database. When a retrieval request is received from a user, a first retrieval section retrieves information from the database using a retrieval expression in the retrieval request. A retrieval result memory section stores the information retrieved by the first retrieval section. If a new retrieval request is not received and detailed information corresponding to the retrieval expression is not stored in the retrieval result memory section, a second retrieval section retrieves more appropriate information from the database using a modified retrieval expression and/or more sophisticated retrieval algorithm.

26 Claims, 10 Drawing Sheets

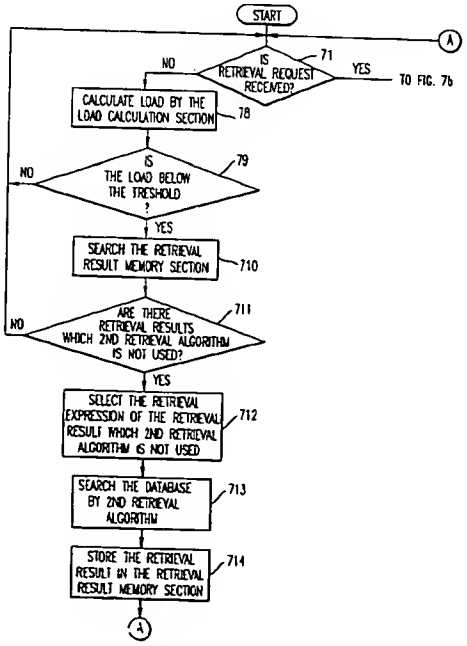
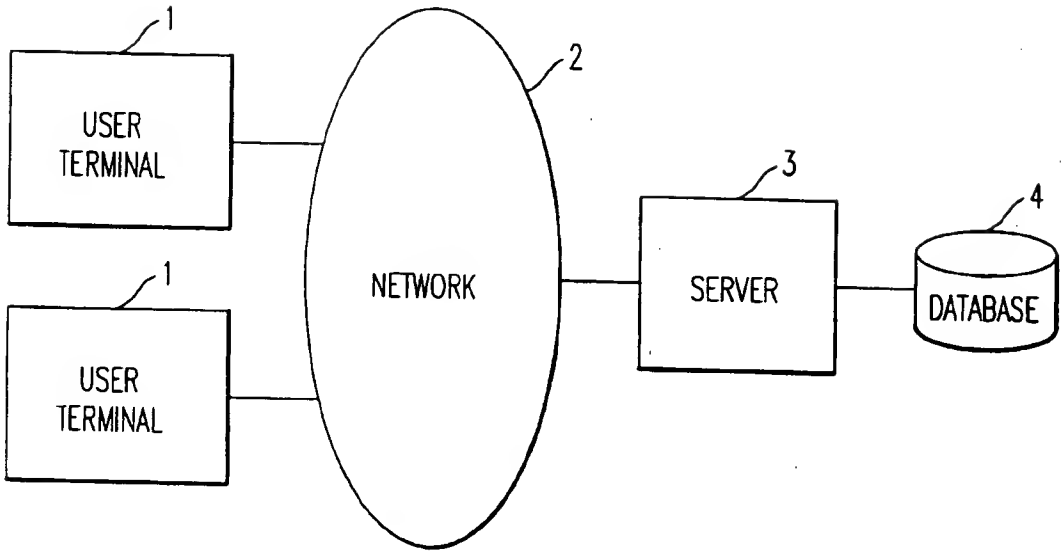


FIG. 1


1
USER TERMINAL
1
USER TERMINAL
2
NETWORK
3
SERVER
4
DATABASE

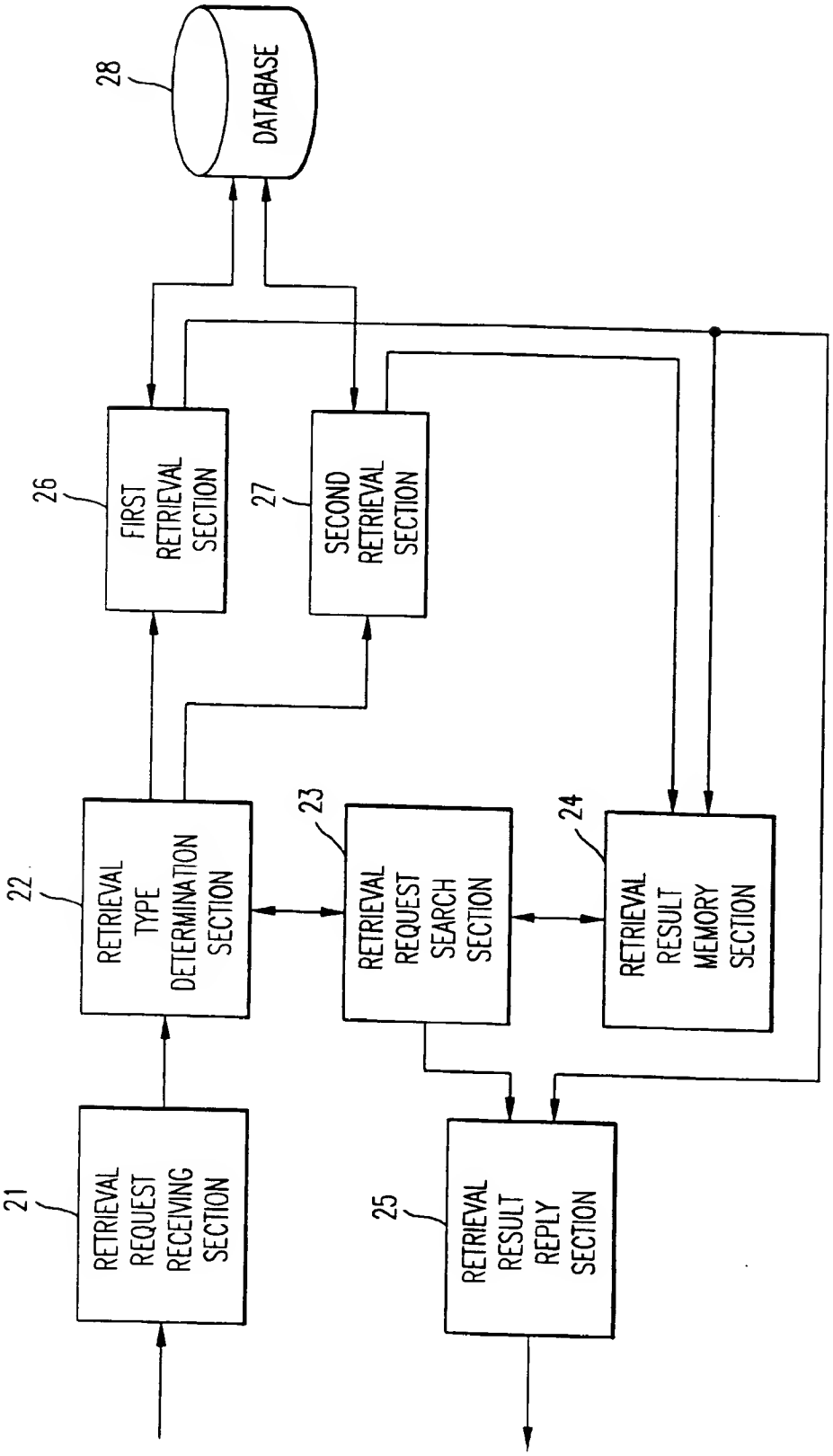

28
DATABASE
26
FIRST RETRIEVAL SECTION
27
SECOND RETRIEVAL SECTION
22
RETRIEVAL TYPE DETERMINATION SECTION
23
RETRIEVAL REQUEST SEARCH SECTION
24
RETRIEVAL RESULT MEMORY SECTION
21
RETRIEVAL REQUEST RECEIVING SECTION
25
RETRIEVAL RESULT REPLY SECTION

FIG. 2

FIG. 3a

31 RETRIEVAL EXPRESSION	32 FIRST RETRIEVAL	33 SECOND RETRIEVAL	34 RETRIEVAL RESULT
AA AND BB	DONE	DONE	... ...
CC AND DD	DONE	DONE	... ...
(BB AND CC) OR (EE AND FF)	DONE		... ...
AA AND BB AND CC	DONE		... ...
CC AND DD AND EE	DONE		... ...
:	:	:	:

FIG. 3b

RETRIEVAL EXPRESSION	FIRST RETRIEVAL	SECOND RETRIEVAL	35 NUMBER OF RETRIEVAL REQUEST	RETRIEVAL RESULT
AA AND BB	DONE	DONE	411	... ...
CC AND DD	DONE		12	... ...
(BB AND CC) OR (EE AND FF)	DONE		48	... ...
AA AND BB AND CC	DONE	DONE	143	... ...
CC AND DD AND EE	DONE	DONE	54	... ...
:	:	:	:	:

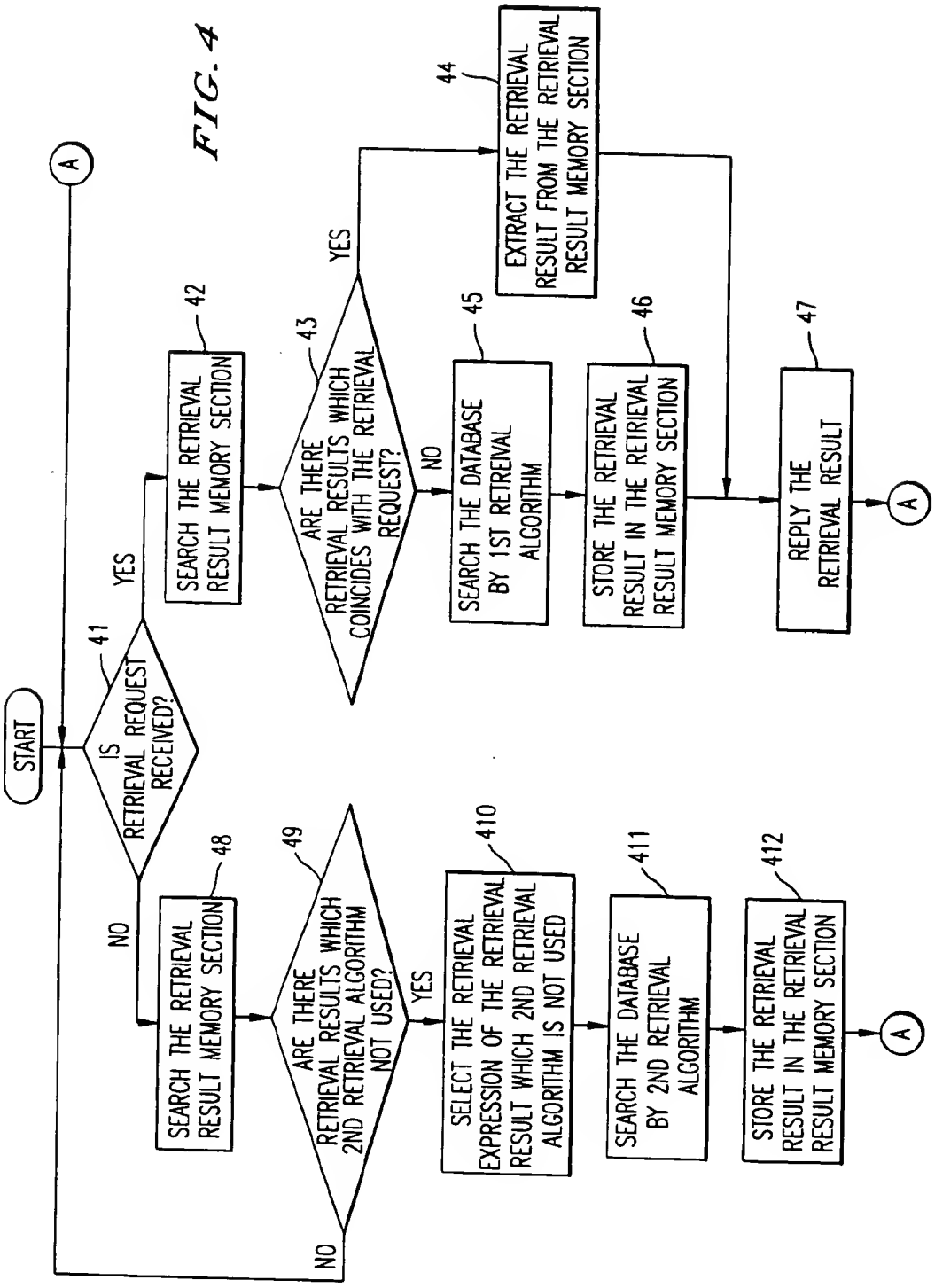

START
IS RETRIEVAL REQUEST RECEIVED?
41
YES
NO
SEARCH THE RETRIEVAL RESULT MEMORY SECTION
42
ARE THERE RETRIEVAL RESULTS WHICH COINCIDES WITH THE RETRIEVAL REQUEST?
43
YES
NO
EXTRACT THE RETRIEVAL RESULT FROM THE RETRIEVAL RESULT MEMORY SECTION
44
SEARCH THE DATABASE BY 1ST RETRIEVAL ALGORITHM
45
STORE THE RETRIEVAL RESULT IN THE RETRIEVAL RESULT MEMORY SECTION
46
REPLY THE RETRIEVAL RESULT
47
A
SEARCH THE RETRIEVAL RESULT MEMORY SECTION
48
ARE THERE RETRIEVAL RESULTS WHICH 2ND RETRIEVAL ALGORITHM NOT USED?
49
YES
NO
SELECT THE RETRIEVAL EXPRESSION OF THE RETRIEVAL RESULT WHICH 2ND RETRIEVAL ALGORITHM IS NOT USED
410
SEARCH THE DATABASE BY 2ND RETRIEVAL ALGORITHM
411
STORE THE RETRIEVAL RESULT IN THE RETRIEVAL RESULT MEMORY SECTION
412
A

FIG. 4

FIG. 5

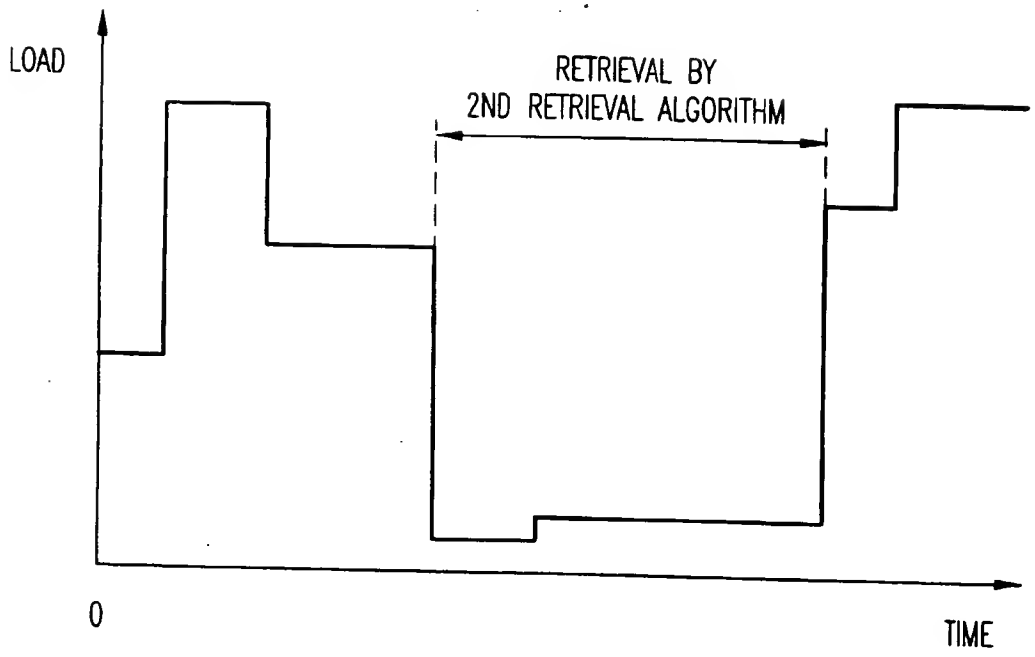


FIG. 6

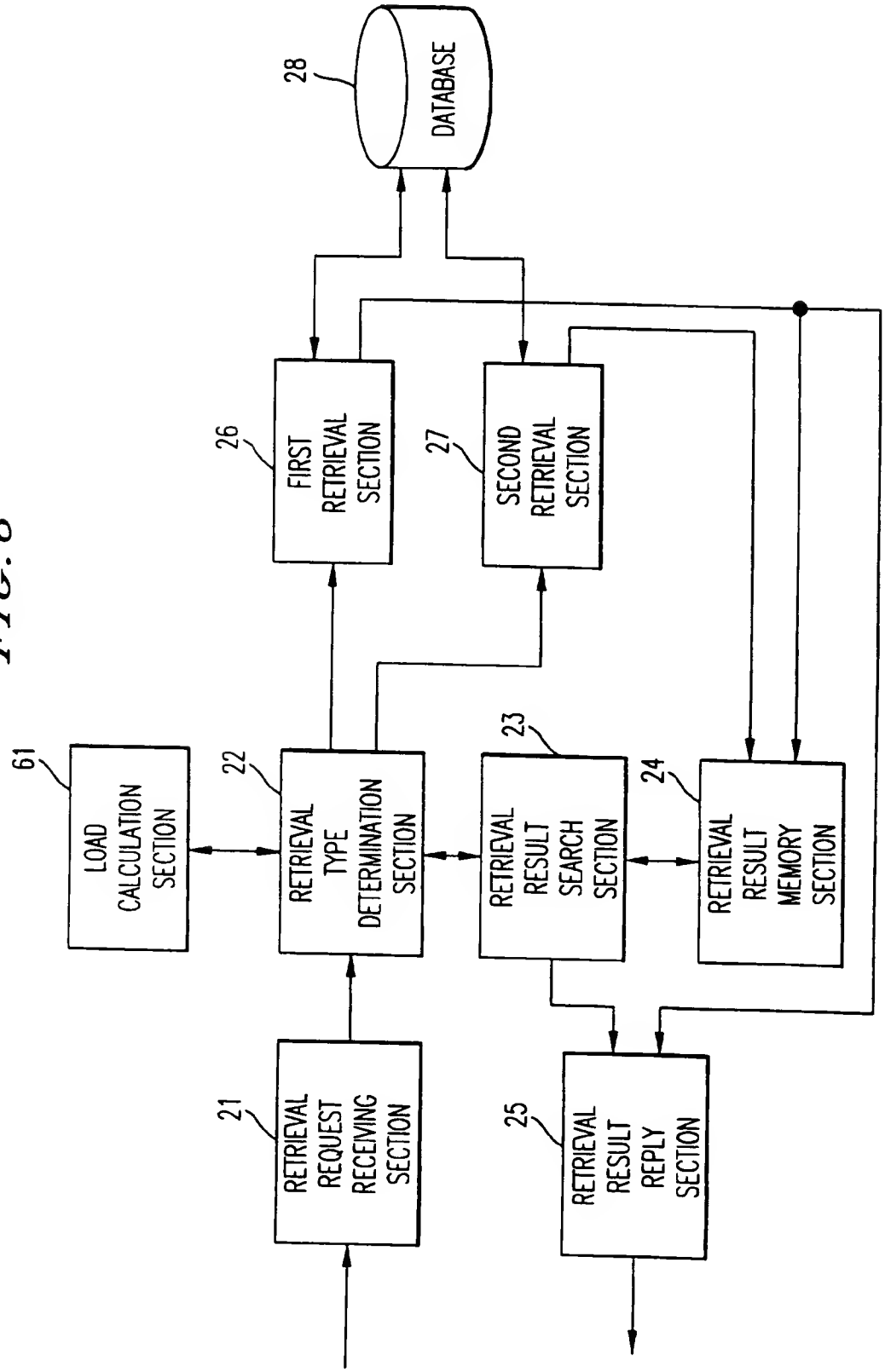

28
DATABASE
26
FIRST RETRIEVAL SECTION
27
SECOND RETRIEVAL SECTION
61
LOAD CALCULATION SECTION
22
RETRIEVAL TYPE DETERMINATION SECTION
23
RETRIEVAL RESULT SEARCH SECTION
24
RETRIEVAL RESULT MEMORY SECTION
21
RETRIEVAL REQUEST RECEIVING SECTION
25
RETRIEVAL RESULT REPLY SECTION

FIG. 7a

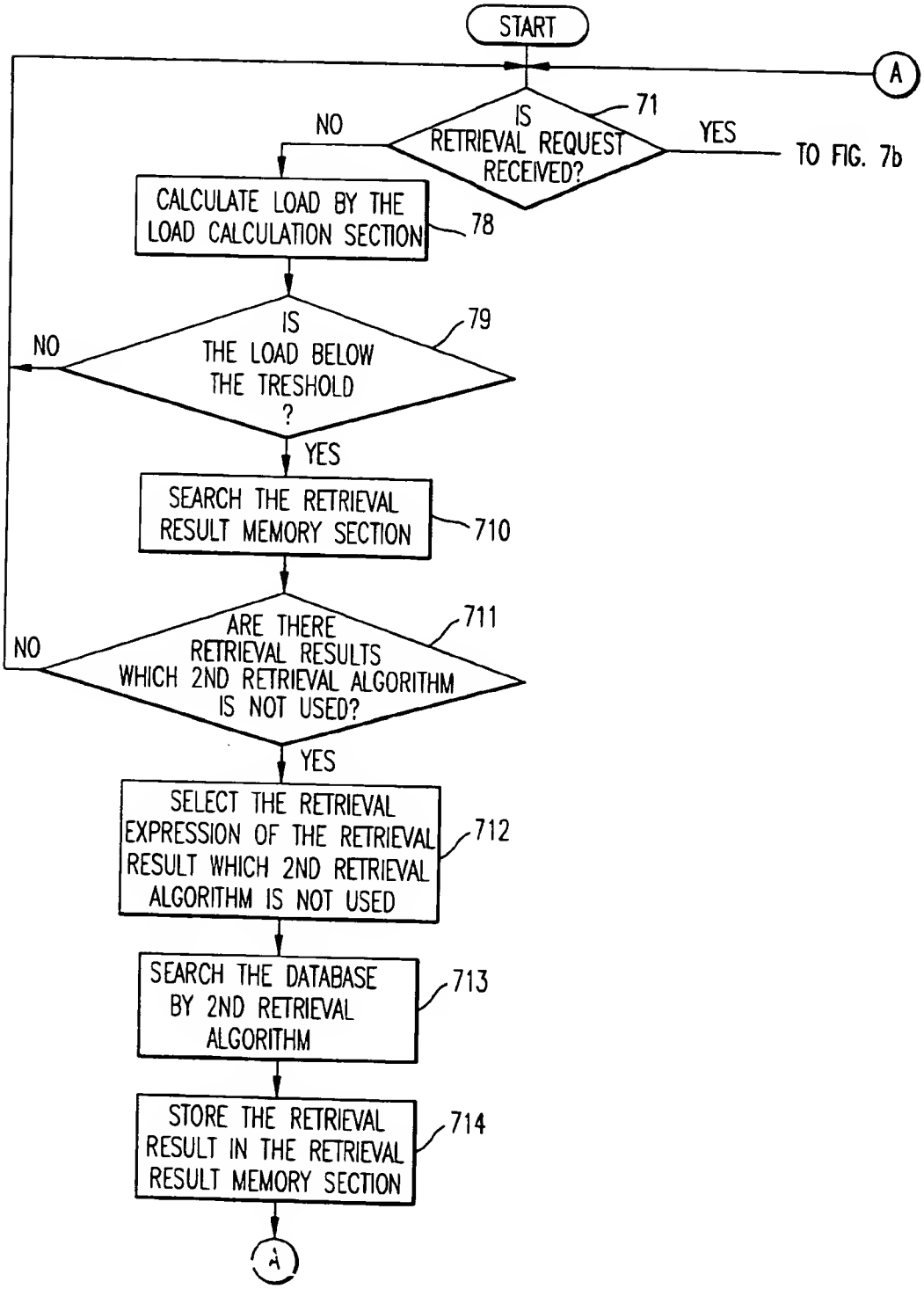
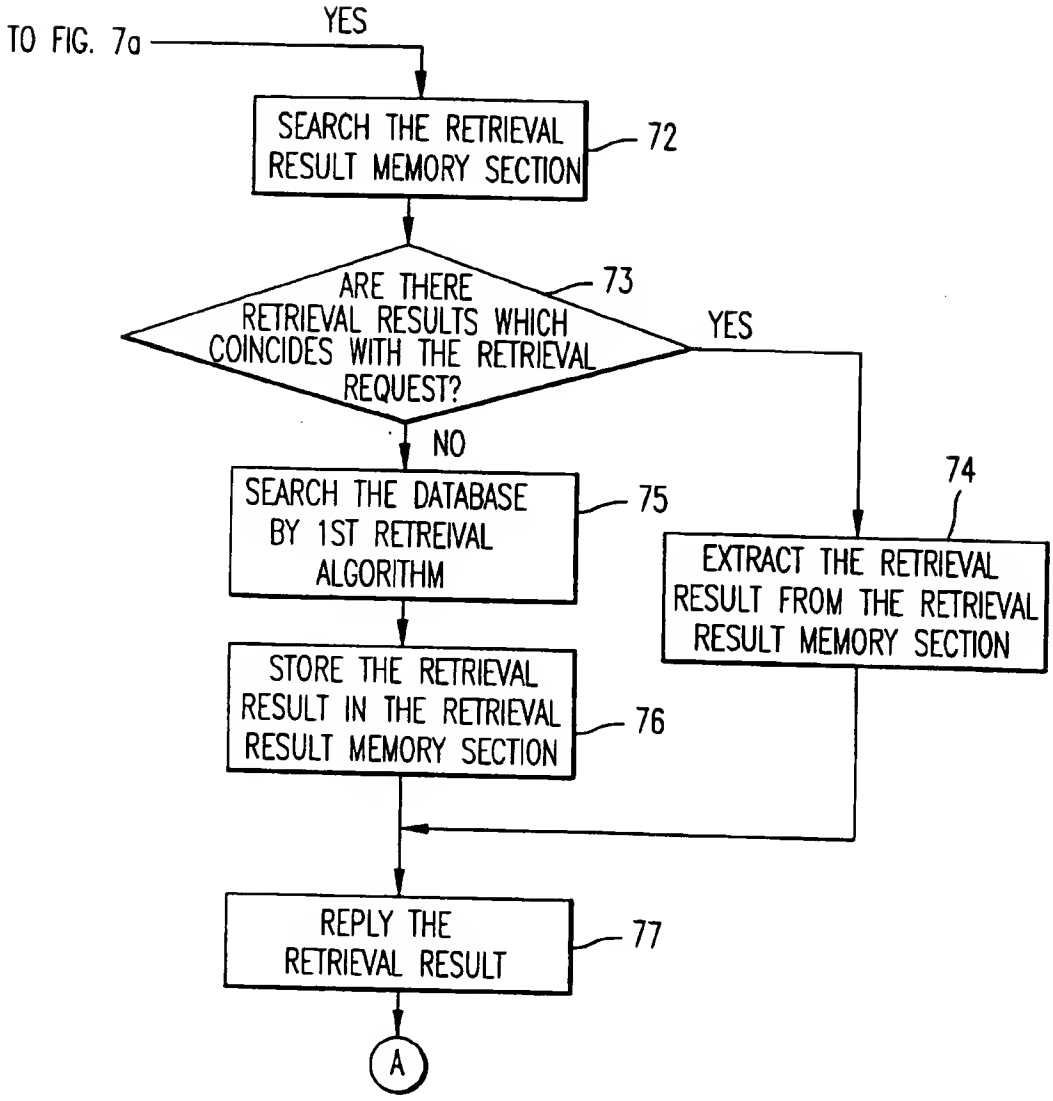

START
A
IS RETRIEVAL REQUEST RECEIVED?
71
NO
YES
TO FIG. 7b
CALCULATE LOAD BY THE LOAD CALCULATION SECTION
78
IS THE LOAD BELOW THE TRESHOLD ?
79
NO
YES
SEARCH THE RETRIEVAL RESULT MEMORY SECTION
710
ARE THERE RETRIEVAL RESULTS WHICH 2ND RETRIEVAL ALGORITHM IS NOT USED?
711
NO
YES
SELECT THE RETRIEVAL EXPRESSION OF THE RETRIEVAL RESULT WHICH 2ND RETRIEVAL ALGORITHM IS NOT USED
712
SEARCH THE DATABASE BY 2ND RETRIEVAL ALGORITHM
713
STORE THE RETRIEVAL RESULT IN THE RETRIEVAL RESULT MEMORY SECTION
714
A

FIG. 7b


TO FIG. 7a
YES
SEARCH THE RETRIEVAL RESULT MEMORY SECTION
72
ARE THERE RETRIEVAL RESULTS WHICH COINCIDES WITH THE RETRIEVAL REQUEST?
73
YES
NO
74
SEARCH THE DATABASE BY 1ST RETREIVAL ALGORITHM
75
EXTRACT THE RETRIEVAL RESULT FROM THE RETRIEVAL RESULT MEMORY SECTION
STORE THE RETRIEVAL RESULT IN THE RETRIEVAL RESULT MEMORY SECTION
76
REPLY THE RETRIEVAL RESULT
77
A

syntax 272 of the index module and search modules utilized. For complex topic areas that are of high priority to the community, the predefined search atoms typically is combined into molecules manually 274. This simulates the action of the query builder for a given profile. Subsequently, the search atoms or search molecules should be iteratively tested to ensure the accuracy of results 276. The lexicon developer can allow a user to choose the high-level topic areas 278 and create or personalize a target profile 280 in order to develop a default target profile. Feedback can be gathered from the community to assist with tbe development or refinement of the community lexicon predefined search queries 282. For ongoing maintenance and improvement, reports from the pattern analysis module that uncover relationships or statistical occurrences typically are examined 284 and used to refine the community lexicon. The reports from the pattern analysis module may also be analyzed for the individualized lexicon free form search queries 286. At this point, if the lexicon can effectively retrieve electronic objects placed on the community's interests and topics, lexicon development is complete 288.

FIG. 12 details the procedure followed by the pattern analysis module when analyzing the electronic objects returned by the searching subsystem. The pattern analysis module first reads 300 metadata, indexes, abstracts, and ratings of the retrieved electronic objects stored in the index table. Statistics are produced 301 describing the frequency and patterns of terms in each object. The pattern analysis module may, more specifically, find the occurrence of lexicon terms 314. The occurrence of lexicon terms in association with other lexicon terms in electronic objects may also be compiled. These data can then be used for many different analysis purposes. The pattern analysis module itself may perform further processing of the data. In tbe alternative, the pattern analysis module may be configured to provide a pipeline of associated terminology for data analysis by other modules that may be added to the system.

One type of data analysis that can be performed by the pattern analysis module or an additional module is to identify improvements that could be made to the lexicon. The electronic objects may be identified based on the frequency 302 of selection. For the electronic object, the underlying search elements 304 are identified and custom search queries are generated 306 which are associated witb the personalized topic area. The pattern analysis module is then able to automatically add 308 tbe custom searcb elements to the user target profile or recommend 310 the custom searcb elements to the system administrator.

A second type of analysis that could be performed by the pattern analysis module is to identify and store 312 the popularity of each object. This allows users to select those topic areas which are most popular within the community.

A typical pattern analysis module also identifies occurrences of patterns 314 of certain lexicon terms. The statistics 316 of the occurrence patterns of the lexicon terms are then stored and can be used to refine the lexicon or be fed to other processing modules.

An example of further processing of tbe frequency of occurrence of terms would be to find a pattern over time in the electronic documents. For example, tbe pattern of merger and acquisition venture activity could be identified over time for a specific company or specific industry. The pattern analysis module, or an added module, may be used to identify the other companies involved in the merger and acquisition ventures, the intensity of the activity, the other industries involved, or other useful information which involves plotting the occurrence of available lexicon terms or user-specified terms.

The various embodiments described above are provided by way of illustration only and should not be construed to limit the invention. Those skilled in the art will readily recognize the various modifications and changes wbich may be made to the present invention without strictly following the exemplary embodiments illustrated and described herein, and without departing from the true spirit and scope of the present invention, which is set forth in the following claims.

What is claimed:

1. A system for search and retrieval of electronic objects, the objects including electronically encoded information, the system comprising:

a searching subsystem comprising

one or more electronic lexicons in a memory within the system, wherein each lexicon is configured to provide predefined search elements designed to identify objects relevant to a specific community; and

a format filter subsystem coupled to the searching subsystem comprising a plurality of format filter modules operable with the lexicon and configured to identify a format of an electronic object and to select a corresponding one of the format filter modules that will enable the system to search the object using the search elements within the lexicon;

a profile management subsystem coupled to the lexicon comprising a community module, a profile module, and an atlas module, wherein the community module is configured to enable selection of a community lexicon, wherein eacb community lexicon includes a library of topics and search elements, wherein the profile module is configured to enable creation of a topic profile by selecting at least one topic from a library of topics, wherein eacb topic identifies a subject that is relevant to the information needs of the community, and wberein tbe atlas module is configured to enable creation of a user atlas by indicating at least one preferred data resource from a list of data resources from which objects may be retrieved;

wbereby potential sources of information can be easily searched and relevant information can be retrieved for a user.

2. The system of claim 1, wherein the searching subsystem further comprises a community module configured to enable selection of a lexicon, wherein each lexicon stores a library of topics and corresponding search elements.

3. The system of claim 2, wherein each topic within the library of topics is associated with one or more of tbe predefined search elements within the lexicon, and wherein each topic identifies a subject that is relevant to the information needs of the community.

4. The system of claim I, wherein the searching subsystem further comprises a profile module configured to enable creation of a target profile by selecting at least one topic from a library of topics, wherein each topic is associated with one or more of the predefined search logic elements and eacb topic identifies a subject or concept of interest that is relevant to the information needs of the community.

5. The system of claim 1, wherein the searcbing subsystem further comprises an atlas module configured to enable creation of a user atlas by selecting at least one preferred data resource from a list of data resources from which objects may be retrieved.

6. The system of claim 1, wherein the searching subsystem further comprises a query builder module which

accesses a target profile, wherein the target profile lists at least one topic from a library of topics, wherein each topic is associated with one or more of the predefined search elements and each topic identifies a subject or concept of interest that is relevant to the information needs of the community, and wherein the query builder module is configured to create an electronic master search query by concatenating the search elements associated with each topic listed in the target profile.

7. The system of claim 6, wherein a master search module is configured to use the electronic master search query to search at least one electronic object within at least one database listed in a user atlas.

8. The system of claim 7, wherein the master search module is scheduled to automatically search for electronic objects at time intervals.

9. The system of claim 1, further comprising a retrieval subsystem comprising a retrieval module configured to select the corresponding one of the format filter modules for each object identified by the searching subsystem and deliver each object to the user in a viewing format.

10. The system of claim 1, further comprising an indexing subsystem comprising an indexing module configured to create an index of each object identified by the searching subsystem by compiling and storing in computer readable medium summary information that identifies the object;

whereby the system can quickly search the index of the indexed object.

11. The system of claim 1, further comprising a pattern analysis subsystem comprising a pattern analysis module configured to parse through the objects identified by the searching subsystem, and recognize and count words within each object that are in the lexicon.

12. The system of claim 1, further comprising a pattern analysis subsystem configured to locate additional terms within the identified objects according to frequency and location of the terms in relation to words within each object that are in the lexicon.

13. The system of claim 1, further comprising a pattern analysis subsystem configured to record a number of times that each object has been retrieved by the system.

14. A system for search and retrieval of electronic objects, the objects including electronically encoded information, the system comprising:

a searching subsystem comprising

one or more electronic lexicons in a memory within the system, wherein each lexicon is configured to provide predefined search logic elements designed to identify objects relevant to a specific community and topic;

a format filter subsystem coupled to the lexicon comprising a plurality of format filter modules operable with the lexicon and configured to identify a format of an electronic object and to select a corresponding one of the format filter modules that will enable the system to search the object using the search elements within the lexicon; and

a profile management subsystem coupled to the lexicon comprising a community module, a profile module, and an atlas module, wherein the community module is configured to enable selection of a community lexicon, wherein each community lexicon includes a library of topics and search elements, wherein the profile module is configured to enable creation of a topic profile by selecting at least one topic from a library of topics, wherein each topic identifies a subject that is relevant to the information needs of the community, and wherein the atlas module is configured to enable creation of a user atlas by indicating at least one preferred data resource from a list of data resources from which objects may be retrieved;

whereby potential sources of information can be easily searched by selecting relevant topics from a community lexicon and relevant information in many formats can be retrieved for a user.

15. The system of claim 14, further comprising an indexing subsystem comprising an indexing module configured to create an index of each object identified by the searching subsystem by compiling and storing in computer readable medium summary information that identifies each object located by the searching subsystem;

whereby the system can quickly search the index of the indexed object.

16. The system of claim 14, further comprising a pattern analysis subsystem comprising a pattern analysis module configured to sift through the objects identified by the searching subsystem, and recognize and count words within each object that are in the lexicon.

17. The system of claim 14, further comprising a pattern analysis subsystem configured to locate additional terms within the identified objects according to frequency and location of the terms in relation to words within each object that are in the lexicon.

18. The system of claim 14, further comprising a pattern analysis subsystem configured to record a number of times that each object has been retrieved by the system.

19. A method for search and retrieval of electronic objects, the objects including electronically encoded information, the method comprising:

identifying a format of an object to be searched;

selecting a format filter module that is configured to enable searching of the object; and

searching the object using predefined search elements found in an electronic lexicon stored in a memory, wherein each lexicon is configured to provide the predefined search elements designed to identify objects relevant to a specific community and topic;

Managing a profile comprising a community module, a profile module, and an atlas module, wherein the community module is configured to enable selection of a community lexicon, wherein each community lexicon includes a library of topics and search elements, wherein the profile module is configured to enable creation of a topic profile by selecting at least one topic from a library of topics, wherein each topic identifies a subject that is relevant to the information needs of the community and wherein the atlas module is configured to enable creation of a user atlas by indicating at least one preferred data resource from a list of data resources from which objects may be retrieved;

whereby potential sources of information can be easily searched and relevant information can be retrieved for a user.

20. The method of claim 19, further comprising retrieving the object identified in the searching step by using the selected format filter module to present the object to the user in a viewing format.

21. The method of claim 19, the method further comprising selecting a community lexicon, wherein each community lexicon includes a library of topics corresponding search elements, wherein each topic within the library of topics is associated with one or more of the predefined

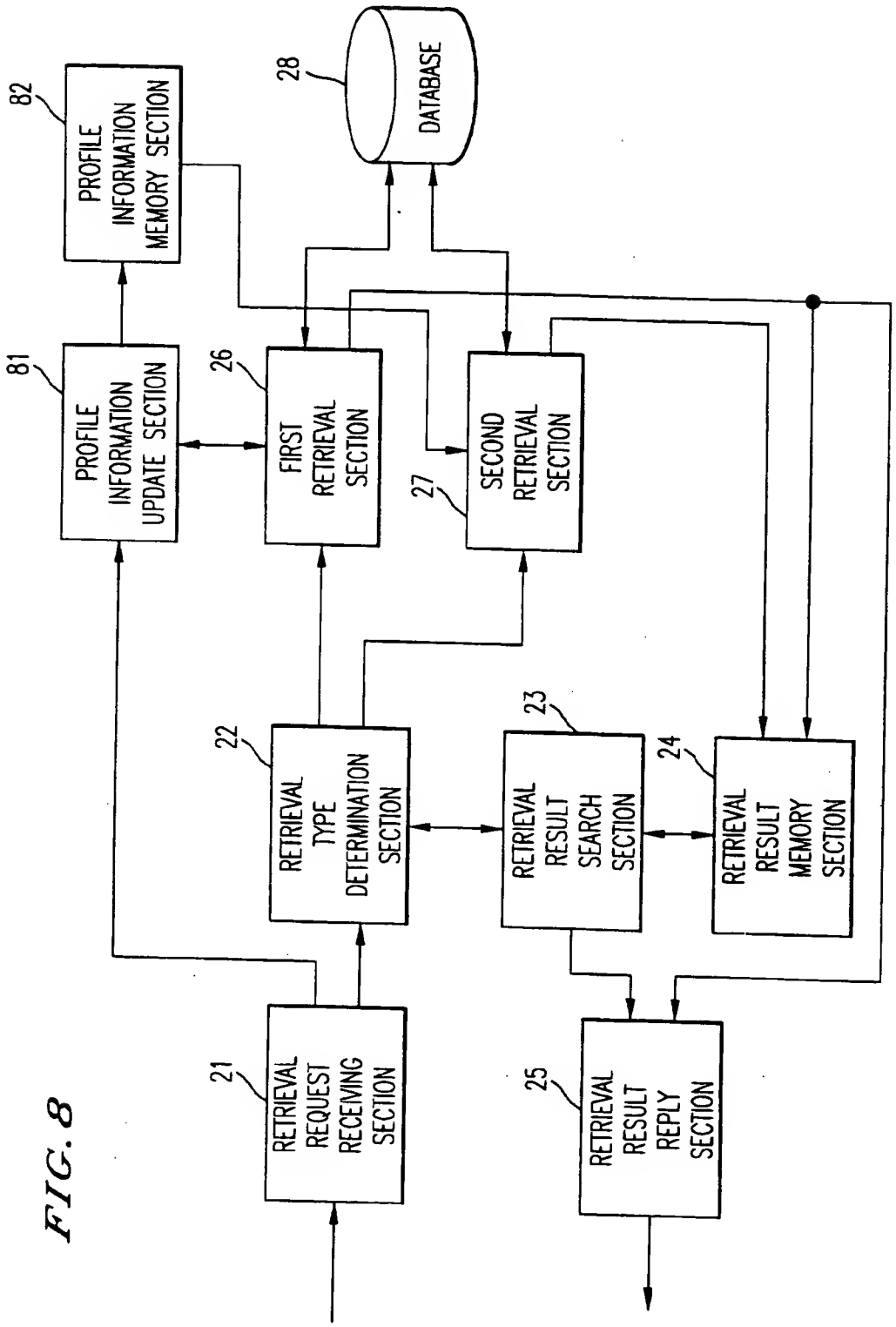

82
PROFILE INFORMATION MEMORY SECTION
28
DATABASE
81
PROFILE INFORMATION UPDATE SECTION
26
FIRST RETRIEVAL SECTION
27
SECOND RETRIEVAL SECTION
22
RETRIEVAL TYPE DETERMINATION SECTION
23
RETRIEVAL RESULT SEARCH SECTION
24
RETRIEVAL RESULT MEMORY SECTION
21
RETRIEVAL REQUEST RECEIVING SECTION
25
RETRIEVAL RESULT REPLY SECTION

FIG. 8

FIG. 9

USER CODE	INTERESTED TOPIC
A	SOCCER
B	PERSONAL COMPUTER
C	ECONOMY
D	SOCCER
E	GOLF
F	SOCCER
G	SOCCER

FIG. 10

INTERESTED TOPIC	RETRIEVAL EXPRESSION	NUMBER OF RETRIEVAL REQUEST
SOCCER	WORLD CUP	10
SOCCER	JAPAN LEAGUE	1
PERSONAL COMPUTER	BROWSER	1

INFORMATION RETRIEVAL APPARATUS AND A METHOD

FIELD OF THE INVENTION

The present invention relates to an information retrieval apparatus and a method to present high quality information to a user quickly in a network system.

BACKGROUND OF THE INVENTION

Recently, the World Wide Web ("WWW") is widely used. In the Internet or Intranet, electric information is easily accessable. Therefore, a so-called information flood occurs and the user can not easily extract his desired data from vast information. In order to reduce this problem, retrieval systems (i.e., search engines) appeared in the Internet or Intranet. As a retrieval system, "Altavista" of DEC and "excite" of Excite corporation are well known. When a retrieval request is received from the user as a retrieval expression, the retrieval system presents the information (Web pages) which best related with the retrieval expression. In this case, a web page as a retrieval object is preserved as a format of an index table and quickly retrieved. However, if a large number of retrieval requests are received at once from many users in the world, it takes a long time to process all of them.

Accordingly, in order to quickly respond to the user for at least one retrieval request, the system caches the previous retrieval result. In this technique, if the information is retrieved from the database by the retrieval expression, the information is stored in a storage device (cache memory) as the retrieval result. Hereafter, when the retrieval request is the same as before, at least one retrieval result is extracted from the storage device and quickly presented to the user. In short, after the second retrieval request, the database is not actually retrieved and the information is provided to the user by referring to the cache content. This technique is disclosed in Japanese Patent Disclosure (Kokai) H4-199468, H4-326163.

However, in this kind of the retrieval system, the quality of the retrieval result is not taken into consideration because a fast retrieval and a high quality of the retrieval result conflict with each other. In short, when an accurate retrieval algorithm improves the quality of the retrieval result, but it takes a long time. On the other hand, in case of the retrieval is executed by a profile (interest topic) by unit of the user, the retrieval results stored in the cache memory is not effective for other users. In the latter case, the response time to present the retrieval result to the user is long because the information is normally retrieved from the database.

Now, characteristics of WWW retrieval in their present condition will be analysed.

Firstly, a huge number of requests come randomly. A large number of the retrieval requests are received through a network, but it is necessary to quickly respond to these retrieval request. Each user respectively locates a terminal connected to the network. Therefore, each retrieval request reaches the retrieval system at random.

Secondly, same retrieval expressions are requested often. A large number of retrieval requests are not different each other. About seventy percent of the retrieval expressions are same. Especially, in the WWW, many users want to know new information. In general, the kind of the new information is limited and the retrieval requests often are the same retrieval expression.

Thirdly, the retrieval with a sophisticated algorithm takes a long time. The retrieval results will be better when the system retrieves the data with synonyms of words in retrieval expression, the integration of character base and word base retrieval, and with the user's profile, but the retrieval time will be long.

As mentioned-above, when using the elaborate retrieval algorithm, the retrieval processing takes a long time. However, if the retrieval results formerly obtained is provided, the retrieval results are poor to the user.

SUMMARY OF THE INVENTION

It is an object of the present invention to provide an information retrieval apparatus and a method to present at least one retrieval result of high quality to the user quickly.

According to the present invention, there is provided an information retrieval apparatus, comprising: a receiving means for receiving a retrieval request from a user; a first retrieval means for retrieving information from a database in response to the retrieval request; a retrieval result memory means for storing the information retrieved by said first retrieval means; and a second retrieval means for retrieving detailed or additional information from the database by using the retrieval request in case a new retrieval request is not received by said receiving means.

Further in accordance with the present invention, there is also provided a method for retrieving information from a database; comprising the steps of: receiving a retrieval request from a user; retrieving the information from the database in response to the retrieval request; storing the information retrieved at the retrieving step in a retrieval result memory; and advanced retrieving information from the database by using the retrieval request in case new retrieval request is not received at the receiving step.

Further in accordance with the present invention, there is also provided a computer readable memory containing computer readable instructions to retrieve information from a database, comprising: an instruction means for causing a computer to receive a retrieval request from a user; an instruction means for causing a computer to retrieve the information from the database in response to the retrieval request; an instruction means for causing a computer to store the retrieved information in a retrieval result memory; and an instruction means for causing a computer to retrieve detail information from the database by using the retrieval request in case new retrieval request is not received.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a block diagram of a retrieval system incorporating an information retrieval apparatus of the present invention.

FIG. **2** is a block diagram of the information retrieval apparatus according to the first embodiment of the present invention.

FIGS. **3A** and **3B** show data stored in a retrieval result memory section in the information retrieval apparatus in FIG. **2**.

FIG. **4** is a flow chart of the processing of the information retrieval method according to the first embodiment of the present invention.

FIG. **5** is a schematic diagram of a time chart of a second retrieval algorithm according to a second embodiment of the present invention.

FIG. **6** is a block diagram of the information retrieval apparatus according to a second embodiment of the present invention.

FIG. **7** is a flow chart of the processing of the information retrieval method according to the second embodiment of the present invention.

FIG. 8 is a block diagram of the information retrieval apparatus according to a third embodiment of the present invention.

FIG. 9 shows interested topic data stored in a profile information memory section in the information retrieval apparatus in FIG. 8.

FIG. 10 shows number of retrieval request for each retrieval expression stored in the profile information memory section in the information retrieval apparatus in FIG. 8.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Embodiments of the present invention will be explained referring to the Figures. FIG. 1 is a block diagram of system incorporating the information retrieval method of the present invention. In FIG. 1, the user inputs the retrieval request from a user terminal 1 to a server 3, which is realized as the information retrieval apparatus of the present invention, through a network 2. The server 3 obtains the retrieval result which coincides with the retrieval request by referring to a large number of data (for example, a table information of Web page) stored in a database 4, and transmits at least one retrieval result to the user terminal 1. In general, as the data stored in the database 4, the Web page is converted to a format of an index table. However, the conversion is not limited to this format. In short, the retrieval object data may be stored by some format.

FIG. 2 is a block diagram of the information retrieval apparatus according to the first embodiment of the present invention. In FIG. 2, when a retrieval request receiving section 21 receives the retrieval request including a predetermined retrieval expression through the network 2, the retrieval request receiving section 21 sends the retrieval request to a retrieval kind determination section 22. The retrieval kind determination section 22 executes latermentioned processing in case the retrieval request is not received from the retrieval request receiving section 21. In case the retrieval request is received, the retrieval kind determination section 22 extracts the retrieval expression from the retrieval request and inquires of a request result search section 23 whether a retrieval by same retrieval expression has been executed in the past. The retrieval result search section 23 examines whether the retrieval result obtained by the same retrieval expression is already stored in a retrieval result memory section 24.

FIGS. 3A and 3B show examples of data stored in the retrieval result memory section 24. As shown in FIG. 3A, by referring to a field 31 of the retrieval expression, the retrieval result search section 23 examines whether the same retrieval expression is already stored. If the same retrieved expression is stored in the field 31, the retrieval result corresponding to the same retrieval expression is extracted from a field 34 of the retrieval result and outputted to a retrieval result reply section 25. On the other hand, if the same retrieval expression is not stored in the field 31, an instruction is sent to the retrieval kind determination section 22 to execute the latermentioned processing. The retrieval result reply section 25 transmits at least one retrieval result received from the retrieval result search section 23 to the user terminal 1 through the network.

In case the retrieval type determination section 22 receives the instruction of non-storing of the retrieval expression from the retrieval result search section 23, the retrieval type determination section 22 requests a retrieval based on the retrieval expression to a first retrieval section 26. The first retrieval section 26 retrieves the information from the database 28 by a fast retrieval algorithm, for example, an algorithm of low retrieval accuracy. The first retrieval section 26 writes the retrieval result in the retrieval result memory section 24 and sends the retrieval result to the retrieval result reply section 25. In this case, when the retrieval result is written in the retrieval result memory section 24, a flag is written in a check field 32 of the first retrieval as shown in FIG. 3A. As an internal expression of the system, for example, the flag "1" is written as the retrieval completion. In FIG. 3A, the flag "done" is shown as the retrieval completion. In the first embodiment, the first retrieval obtained by using the retrieval expression represents the retrieval based on the first retrieval algorithm (method). In case some data exists in the field 31 of the retrieval expression, the flag is also written in the check field 32 of the first retrieval corresponding to the field 31. Therefore, the check field 32 may be omitted by referring to the retrieval expression field 31 and the retrieval result field 34.

In case the retrieval type determination section 22 does not receive the retrieval request from the retrieval request receiving section 21, the retrieval type determination section 22 inquires of the retrieval result search section 23 whether the retrieval expression, whose retrieval result is obtained by the first retrieval algorithm and not obtained by the second retrieval algorithm, is stored in the retrieval result memory section 24. In response to the inquiry, the retrieval result search section 23 extracts the retrieval expression by referring to the retrieval result memory section 24 and transmits the retrieval expression to the retrieval type determination section 22. In this case, as shown in FIG. 3A, the retrieval expression, of which the check field 32 of the first algorithm is written by "done" and the check field 33 of the second algorithm is not written, is extracted. If a plurality of the retrieval expressions are extracted, one retrieval expression is selected by a predetermined standard. As the predetermined standard, for example, the latest retrieval expression is selected. Otherwise, as shown in FIG. 3B, the number of retrieval requests by the retrieval expression are counted in a field 35 and the retrieval expression whose number of retrieval requests is largest is selected. In FIG. 3B, the retrieval expression "(BB and CC) or (EE and FF)" is selected because the number of retrieval requests "48" is the largest value in the retrieval expressions not having the second retrieval flag "done".

When the retrieval type determination section 22 receives the retrieval expression from the retrieval result search section 23, the retrieval type determination section 22 requests a second retrieval section 27 to retrieve a further information by using the retrieval expression. The second retrieval section 27 retrieves the detail information from a database 28 by the second retrieval algorithm whose retrieval accuracy is higher than the first retrieval algorithm, and writes the detail information in the retrieval result memory section 24. In this case, as shown in FIG. 3A, a flag "done" is written in the check field 33.

FIG. 4 is a flow chart of the processing of the information retrieval method according to the first embodiment. First, the retrieval type determination section 22 decides whether the retrieval request is received (step 41). When receiving the retrieval request, the retrieval type determination section 22 sends the instruction of receiving to the retrieval result search section 23. The retrieval result search section 23 retrieves the retrieval result memory section 24 (step 42), and decides whether the retrieval result corresponding to the retrieval expression in the retrieval request is stored in the

retrieval result memory section **24**. (step **43**). When the retrieval result is already stored, the retrieval result search section **23** extracts the retrieval result from the retrieval result memory section **24** (step **44**) and sends it to the retrieval result reply section **25**. Otherwise, the retrieval result search section **23** sends the instruction to the retrieval type determination section **22**. The retrieval type determination section **22** requests the first retrieval section **26** to retrieve hy the first algorithm. The first retrieval section **26** retrieves the datahase **28** by the first algorithm (step **45**), writes the retrieval result in the retrieval result memory section **24**, and sends it to the retrieval result reply section **25** (step **46**). When the retrieval result reply section **25** receives the retrieval result from the retrieval result search section **23** or the first retrieval section **26**, the retrieval result is sent to the user (step **47**).

On the other hand, when the retrieval type determination section **22** does not receive the retrieval request, the instruction of non-receiving is sent to the retrieval result search section **23**. The retrieval result search section **23** searches the retrieval expression, whose retrieval result is obtained hy the first algorithm hut not ohtained hy the second algorithm from the retrieval result memory section **24** (step **48**). If the retrieval expression is searched (step **49**), one retrieval expression is selected by the predetermined standard (step **410**) and outputted to the retrieval type determination section **22**. The retrieval type determination section **22** requests the second retrieval section **27** to retrieve by the second algorithm. The second retrieval section **27** retrieves more appropriate information from the datahase hy the second algorithm (step **411**), and writes the retrieval result to the retrieval result memory section **24**. The flag "done" is written in the check field **33** in the retrieval result memory section **24**.

For example, the first retrieval algorithm retrieves information for the past one year from the database, and the second retrieval algorithm retrieves information for the past ten years. As another example, the first retrieval algorithm retrieves using a keyword (retrieval expression) only and the second retrieval algorithm retrieves hy using not only the keyword but also a synonym or a same category word.

As mentioned-above, if the retrieval request is received and a system load hecomes high, the first retrieval algorithm whose accuracy is low is quickly executed. In this case, the response time to the user is short. On the other hand, while another retrieval request is not received, the second retrieval algorithm whose accuracy is high is executed. In this case, the detailed information for the retrieval request received in the near future is previously stored. As a result, in response to the retrieval request in the future, the retrieval result of high quality is quickly presented to the user.

In the first emhodiment, while the retrieval request is not received, the second retrieval algorithm is executed. However, in the second emhodiment, by monitoring the load of the system in addition to non-receiving of the retrieval request, the second retrieval algorithm is decided to be executed. FIG. **5** is a time chart showing a change status of the system load. As shown in FIG. **5**, while the system load is low, the second retrieval algorithm is executed.

FIG. **6** is a block diagram of the information retrieval apparatus according to the second emhodiment. In comparison with the first embodiment in FIG. **2**, a load calculation section **61** differs in the hlock diagram. In case the retrieval request is not received, the retrieval type determination section **22** asks the load calculation section **61** for the load status of the system. The load calculation section **61** calculates the load value of the system at the present time, and sends the load value to the retrieval type determination section **22**. The load value is calculated by using a system call of the operating system or a load average of the CPU. The load value may be the average value at predetermined interval. When the retrieval type determination section **22** receives the load value, the retrieval type determination section **22** decides whether the load value is below a predetermined threshold. If the load value is below the predetermined threshold (the system load is small), the retrieval type determination section **22** asks the retrieval result search section **23** to search the retrieval expression whose retrieval result is ohtained hy the first algorithm hut not ohtained hy the second algorithm from the retrieval result memory section **24**. Hereafter, processing is same as the first embodiment. Therefore, explanation is omitted.

FIG. **7** is a flow chart of processing of the information retrieval method according to the second emhodiment. In comparison with the flow chart of the first embodiment in FIG. **4**, steps **78** and **79** are different steps in the flow chart. In FIG. **7**, in case the retrieval request is not received (step **71**), the retrieval type determination section **22** asks the load calculation section **61** for the load value. The load calculation section **61** calculates the load value (step **78**). The retrieval type determination section **22** compares the load value with the predetermined threshold. If the load value is helow the predetermined threshold (step **79**), processing from step **710** is continuously executed.

As mentioned-above, in the second emhodiment, the actual load of the system is monitored while the retrieval request is not received. Accordingly, if the load of the system is high for some reason, the processing to highly raise the load (second retrieval algorithm) is avoided. As a result, the system is effectively used. In case the predetermined server **3** in FIG. **1** is utilized as not only the retrieval system but also as various kinds of processing systems, the second emhodiment is especially useful.

In the first and second embodiments, while another retrieval request is not received and so on, the second retrieval algorithm is executed for the retrieval expression whose retrieval result is not ohtained by the first retrieval algorithm. However, in the third emhodiment, the retrieval algorithm is executed according to a user's attribute profile in addition to the retrieval expression. For example, assume that the first retrieval algorithm and the second retrieval algorithm are already executed using a keyword "personal computer" in the retrieval request. In this case, if many users are interested in "Economy", the detail information is previously retrieved hy both the keyword "Personal Computer" and the keyword "Economy".

FIG. **8** is a block diagram of the information retrieval apparatus according to the third emhodiment. In comparison with the first emhodiment, a profile information update section **81** and a profile information memory section **82** differ in the block diagram. First, when the retrieval request including the retrieval expression S is received from a user A, the retrieval request receiving section **21** sends the instruction to the profile information update section **81**. If an area of the user A is not secured in the profile information memory section **82**, the profile information update section **81** secures an area for the user A in the profile information memory section **82**. The first retrieval section **26** or the second retrieval section **27** extracts the predetermined element (for example, a keyword) from the retrieval expression S, and sends the keyword to the profile information update section **81**. The profile information update section **81** writes the keyword in the area of the user A of the profile infor-

mation memory section **82**. In this case, the profile information memory section **82** may store the number of appearance frequency by unit of the keyword. In short, the retrieval corresponding to a different retrieval request by unit of the user may be executed.

Next, assume that the first retrieval section **26** retrieves data from the database, stores n-units of high rank of the first retrieval results in the retrieval result memory section **24**, and sends m-units (m<n) of high rank from the n-units retrieval results to the user A. Then, if the second retrieval section **27** executes the second retrieval algorithm by using the retrieval expression S in the same way as the first embodiment, n-units of high rank of the second retrieval results are also stored in the retrieval result memory section **24**. However, in this case, the n-units of high rank of the first retrieval results are not deleted in the retrieval result memory section **24**. The n-units of high rank of the first retrieval results are irrelevant to the second retrieval results from (m+1)th-unit of high rank when the user A wants to watch the first retrieval results from (m+1)th-unit of high rank. In order to avoid this situation, in the case of the "WWW", the user is identified by using a "cookie" or "fat URL". As for the user A, the first retrieval results from (m+1)th-unit of high rank is presented. In this case, the storing time of the first retrieval results in the retrieval result memory section **24** is previously determined. When the storing time passed, the first retrieval results are deleted in the retrieval result memory section **24**. Accordingly, if another user inputs the retrieval request including the retrieval expression S, the second retrieval results will be presented to the other user.

Next, a modification of the third embodiment is explained. FIG. **9** shows a table of interested topic for each user stored in the profile information memory section **82**. First, each user previously registers his interested topic to the profile information memory section **82** through the user terminal **1**. Hereafter, whenever each user inputs the retrieval request including the retrieval expression, the profile information update section **81** counts the number of the retrieval request by unit of the retrieval expression for each interest topic. FIG. **10** shows a table of the counted number of retrieval requests by unit of the retrieval expression for each interest topic stored in the profile information memory section **82**. In this way, when the retrieval request is not received, the second retrieval section **27** retrieves the detail information by using the retrieval expression with the largest number of retrieval requests. In FIG. **10**, by using the keywords "SOCCER" and "WORLD CUP", the second retrieval algorithm is executed. Accordingly, the detail information in which many users are interested at the present time is previously stored, and quickly presented to the user when he inputs the retrieval request including this retrieval expression.

In case a quantity of information (for example, the document) as retrieval objects increases, the content of the database **28** is updated in proportion to the increased quantity of the information. In this case, the updated database **28** often affects the retrieval results stored in the retrieval result memory section **24**. For example, in case the retrieval results by a keyword "Personal Computer" is already stored in the retrieval result memory section **24** and a new document including the word "Personal Computer" is added to the database **28**, the retrieval results by the keyword "Personal Computer" must be deleted in the retrieval result memory section **24**. Therefore, in the fourth embodiment, if the new document is added to the database, the word included in the new document is determined to coincide with the retrieval expression in the retrieval result memory section **24**. If the word coincides with the retrieval expression, the retrieval results are deleted in the retrieval result memory section **24**. Furthermore, if the document is deleted from the database, the retrieval results including at least one part of the document, is determined to be stored in the retrieval result memory section **24**. If the retrieval results, including at least one part of the document, are already stored, the retrieval results are deleted in the retrieval result memory section **24**. As for the deleted retrieval results, it is possible for the user to retrieve again. Therefore, while the load of the retrieval engine is low, retrieval for the updated database is executed again by using same retrieval expression. In this case, if the retrieval results corresponding to a plurality of the retrieval expressions are deleted, the retrieval expression is selected in priority order to retrieve the updated database. For example, the retrieval expression of the largest number of retrieval requests, or the retrieval expression of the most recent use is selected.

In the first, second, and third embodiments, the first retrieval algorithm and the second retrieval algorithm are used. However, a plurality of different kinds of retrieval algorithms, i.e., more than two, may be used. For example, after the second retrieval algorithm is executed, the third retrieval algorithm whose accuracy is higher than the second retrieval algorithm may be executed during idle time of the computer. By repeating this processing, arbitrary steps of the retrieval algorithm whose accuracy is higher in order are used.

Furthermore, as an example in the above-mentioned embodiment, the retrieval is executed for text data in the database. However, the retrieval may be executed for multimedia data in the database. For example, as for an image retrieval algorithm of low accuracy, a matching as a histogram level is executed by using a reduced image or a low resolution image. As for an image retrieval algorithm of high accuracy, a matching of high level is executed by using a high resolution image.

A memory can be used to store instructions for performing the process of the present invention described above. Such a memory can be a hard disk, optical disk, semiconductor memory, and so on.

Other embodiments of the invention will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein. It is intended that the specification and examples be considered as exemplary only, with the true scope and spirit of the invention being indicated by the following claims.

What is claimed is:

1. An information retrieval apparatus, comprising:

a receiving unit configured to receive from a user, retrieval requests including a first retrieval request;

a first retrieval unit configured to retrieve first information from a database in response to the first retrieval request, in accordance with a first retrieval algorithm that has a first retrieval accuracy;

a retrieval result memory configured to store the first retrieval request and the first information;

a retrieval result reply unit configured to send the first information to the user; and

a second retrieval unit configured to use the stored first retrieval request to retrieve from the database, while a load of the apparatus is below a threshold, high quality information that has a higher quality than the first information, in accordance with a second retrieval algorithm that has a second retrieval accuracy that is higher than the first retrieval accuracy;

wherein said retrieval result memory stores the high quality information retrieved by said second retrieval unit; and

wherein, if said receiving unit receives a second retrieval request that is identical to or similar to the stored first retrieval request, said retrieval result reply unit sends the high quality information stored in said retrieval result memory to a user of the second retrieval request.

2. The information retrieval apparatus of claim 1, wherein:

said second retrieval unit retrieves the high quality information from the database while said first retrieval unit does not process the first information in response to the first retrieval request.

3. The information retrieval apparatus of claim 1, further comprising:

a load calculation unit that calculates the load of the apparatus.

4. The information retrieval apparatus of claim 1, further comprising:

a profile information memory configured to previously store an interested topic for each user code, and to count a number of retrieval requests for each retrieval expression in each interested topic.

5. The information retrieval apparatus of claim 4, wherein:

said second retrieval unit retrieves the high quality information from the database by both the interested topic and the retrieval expression having the highest number of retrieval requests.

6. The information retrieval apparatus of claim 1, wherein:

said first retrieval unit retrieves the first information by at least one retrieval expression in the first retrieval request, and

said second retrieval unit retrieves the high quality information by a modified retrieval expression of the at least one retrieval expression.

7. The information retrieval apparatus of claim 6, wherein:

said retrieval result memory stores the high quality information retrieved by said second retrieval unit, for each retrieval expression.

8. The information retrieval apparatus of claim 7, wherein:

said second retrieval unit retrieves the high quality information from the database by the modified retrieval expression if the high quality information corresponding to the retrieval expression is not stored in said retrieval result memory.

9. The information retrieval apparatus of claim 8, wherein:

said retrieval result reply unit sends the high quality information to the user in response to a retrieval request if the high quality information corresponding to the retrieval expression in the retrieval request is stored in said retrieval result memory.

10. The information retrieval apparatus of claim 7,

wherein said retrieval result memory additionally stores a number of retrieval requests with the information for each retrieval expression.

11. The information retrieval apparatus of claim 10,

wherein said second retrieval unit retrieves the high quality information from the database by the modified retrieval expression of one retrieval expression having the highest number of retrieval requests in a plurality of retrieval expressions if high quality information corresponding to the plurality of retrieval expressions is not stored in said retrieval result memory.

12. A method for retrieving information from a database, comprising:

receiving a first retrieval request from a user;

first-retrieving first information from the database in response to the first retrieval request, according to a first retrieval algorithm that has a first retrieval accuracy;

storing in a retrieval result memory, the first information and the first retrieval request;

sending the first information to the user;

while a system load is below a threshold, using the stored first retrieval request in a step of second-retrieving from the database, high quality information that has a higher quality than the first information, in accordance with a second retrieval algorithm that has a second retrieval accuracy that is higher than the first retrieval accuracy;

storing in the retrieval result memory, the high quality information retrieved in the second-retrieving step; and

if a second retrieval request is received that is identical to or similar to the stored first retrieval request, sending the high quality information to a user of the second retrieval request.

13. The method for retrieving information of claim 12, wherein:

the first information is retrieved at the first-retrieving step by a retrieval expression in the first retrieval request; and

the high quality information is retrieved at the second-retrieving step by a modified retrieval expression of the retrieval expression.

14. The method for retrieving information of claim 12, wherein:

the first information is retrieved at the first-retrieving step by a retrieval expression in the first retrieval request; and

wherein the high quality information is retrieved at the second-retrieving step with integrated algorithm of word-base search and character-base search.

15. The method for retrieving information of claim 12, wherein:

the first information is retrieved at the first-retrieving step by a retrieval expression in the first retrieval request; and

the high quality information is retrieved at the second-retrieving step from the database including more information in terms of time from older to newer.

16. The method for retrieving information of claim 12, wherein:

the high quality information is retrieved at the second-retrieving step while the first information is not processed in response to the first retrieval request.

17. The method for retrieving information of claim 12, further comprising:

storing an interested topic for each user code; and

counting a number of retrieval requests for each retrieval expression in each interested topic.

18. The method for retrieving information of claim 17, wherein:

the high quality information is retrieved at the second-retrieving step by both the interested topic and the

retrieval expression having the highest number of retrieval requests.

19. The method for retrieving information of claim 12, further comprising:

calculating the system load.

20. The method for retrieving information of claim 13, wherein:

the high quality information is stored at the storing step in the retrieval result memory for each retrieval expression.

21. The method for retrieving information of claim 20, wherein:

the high quality information is retrieved at the second-retrieving step by the modified retrieval expression if the high quality information corresponding to the retrieval expression is not stored in the retrieval result memory.

22. The method for retrieving information of claim 21, wherein:

the high quality information is sent at the sending step to the user in response to the retrieval request if the high quality information corresponding to the retrieval expression in the retrieval request is stored in the retrieval result memory.

23. The method for retrieving information of claim 20, further comprising:

storing a number of retrieval requests with the information retrieved at the first-retrieving step for each retrieval expression.

24. The method for retrieving information of claim 23, wherein:

the high quality information is retrieved at the second-retrieving step by the modified retrieval expression of one retrieval expression having the highest number of retrieval requests in a plurality of retrieval expressions if high quality information corresponding to the plurality of retrieval expressions is not stored in the retrieval result memory.

25. A computer readable memory containing computer readable instructions to retrieve information from a database, the computer readable memory comprising:

instruction means for causing a computer to receive a first retrieval request from a user;

instruction means for causing the computer to retrieve first information from the database in response to the first retrieval request, in accordance with a first retrieval algorithm that has a first retrieval accuracy;

instruction means for causing the computer to store in a retrieval result memory, the first information and the first retrieval request;

instruction means for causing the computer to send the first information to the user;

instruction means for causing the computer to use the stored first retrieval request to retrieve from the database, while a system load is below a threshold, high quality information that has a higher quality than the first information, according to a second retrieval algorithm that has a second retrieval accuracy that is higher than the first retrieval accuracy;

instruction means for causing the computer to store in the retrieval result memory, the high quality information retrieved according to the second retrieval algorithm; and

instruction means for causing the computer, if a second retrieval request is received that is identical to or similar to the stored first retrieval request, to send the high quality information to a user of the second retrieval request.

26. The computer readable memory of claim 25, further comprising:

instructions means for causing the computer to calculate the system load.

* * * * *